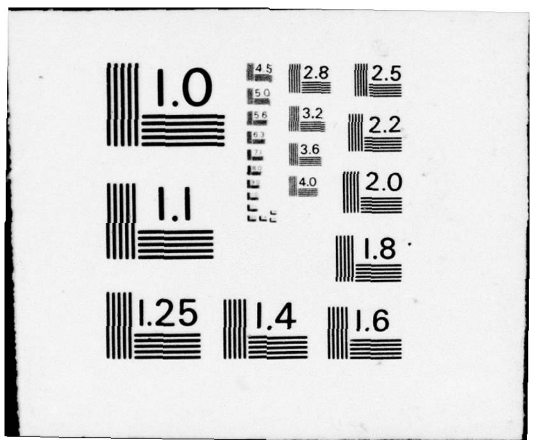
AD-A032 719 NAVAL ACADEMY ANNAPOLIS MD F/G 9/3
MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER ANALYSIS OF TIME --ETC(U)
MAY 76 S J RAHER
UNCLASSIFIED USNA-TSPR-81 NL

1 OF 2
ADA032719


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

AD A032719

27

A TRIDENT SCHOLAR PROJECT REPORT

NO. 81

MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER ANALYSIS OF TIME SERIES DATA

DDC
DEC 2 1976
C

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND
1976

This document has been approved for public release and sale; its distribution is unlimited.

COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

United States Naval Academy - Trident Scholar project report, no. 81 (1976)

6 MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER

ANALYSIS OF TIME SERIES DATA.

9 Research rept.,

21 Report on A Trident Scholar Project Report

by

10 J.

Midshipman Steven Raher, 1/c

U. S. Naval Academy

Annapolis, Maryland

14 USNA-TSPR-81

ACCESSION for

NTIS White Section ☒

DDC Buff Section ☐

UNANNOUNCED ☐

JUSTIFICATION

BY

DISTRIBUTION/AVAILABILITY CODES

Dist. AVAIL. and/or SPECIAL

A

John S. Kalme

Assoc. Professor John S. Kalme
Mathematics Department

Accepted for Trident Scholar Committee

[signature]

Chairman

11 17 MAY 76

Date

12 159 p.

245 600

bpg

MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER ANALYSIS OF TIME SERIES DATA

The purpose of this study is to develop a series of computer programs for use in analyzing time series data (waves), such as EEG readings. Programs were used to produce reliable estimates of correlations, spectra, cross spectra, and partial coherences of multi-channel random processes. The software package was written to be easily adaptable to different sampling rates, amounts of data, and numbers of channels. Provisions for digital prefiltering of data, detrending, and smoothing (using a number of lag windows) were also included. Techniques for estimation of spectra by fitting single- and multi-channel autoregressive schemes to sampled data were also applied and found to yield results consistent with the other methods. All programs were written in FORTRAN and run on the USNA/DTSS computer system.

PREFACE

This study was undertaken as part of a Trident Scholar Research project. It is the result of two semesters of study during the academic year 1975-76. The many hours which my advisor, Assoc. Prof. John S. Kalme, spent contributing help and guidance are sincerely appreciated. I would also like to thank Assoc. Prof. Karel Montor for supplying EEG data, and Maj. David A. Wright (CAF) for his assistance in digitizing the data.

TABLE OF CONTENTS

CHAPTER 1

I. Elementary Definitions

A sample space Ω is the set of all possible outcomes of an experiment. Each possible outcome ω is called an elementary event. A (non-elementary) event A in Ω is any subset of Ω, any collection of elementary events. A probability measure P defined on Ω is a rule which to each event A in Ω assigns a real number P(A) (called the probability of A) such that the following conditions are satisfied:

(1) $P(\emptyset) = 0$

(2) $P(\Omega) = 0$

(3) If A_i are pairwise disjoint events, then $P(\bigcup_i A_i) = \sum_i P(A_i)$

A random variable X is a function defined on a sample space Ω, which assigns a real or complex value to each elementary event ω. The expectation E(X) of the random variable is defined as

$$\int_\Omega X(\omega)\, dP(\omega) \qquad \text{provided} \qquad \int_\Omega |X(\omega)|\, dP(\omega) < \infty$$

Both integrations are performed in a Lebesgue sense.

A random process is a function of two variables, t and ω, where t is a real number or integer, and ω is an elementary event in the sample space Ω. Thus, $X(t,\omega)$ is a random process if t is allowed to vary over an interval, but $X(t_0,\omega)$ for a fixed t_0 is a random variable. Usually the second argument is omitted when expressing a random variable: $X(t,\omega)$ is written as $X(t)$.

A random process is said to be stationary in the wide sense if $E(X(t)\cdot X(t + v))$ depends only upon v, not t. This expectation is called the correlation function of X and is denoted by $R_X(v)$.

If Y(t) is another stationary process defined on the same

sample space, and $E(X(t) \cdot Y(t + v))$ depends only on v, then we say that X and Y are jointly stationary. This expectation is denoted by $R_{XY}(v)$ and is called the cross correlation of the two random processes X and Y.

The Fourier integral of the autocorrelation function of X is called the power spectral density of X, and is evaluated by this expression:

$$S_X(f) = \int_{-\infty}^{\infty} R_X(v)\, e^{-i 2\pi v f}\, dv$$

The power spectral density of a process indicates the amount of energy the process contains in any frequency interval. For example, EEG waveforms have much of their energy concentrated near 10 Hz; if such a wave were passed through a 10 Hz bandpass filter it would lose relatively little power. Thus, we w[illegible] expect its spectral density function to have a peak about 10 Hz.

Similarly, the Fourier intrgral of the cross correlation of X and Y is called the cross spectral density of X and Y and is denoted by $S_{XY}(f)$:

$$S_{XY}(f) = \int_{-\infty}^{\infty} R_{XY}(v)\, e^{-i 2\pi v f}\, dv$$

The cross spectral density represents the amount of power shared by the two processes at any frequency. For example, the cross spectral density of the input and the output of a bandpass filter would have a very high cross spectral density over the frequencies passed by the filter.

Of course, the autocorrelation and cross correlation functions can be recovered from the spectral density and cross spectral density functions, respectively, by using the inverse Fourier

integrals.

If the processes X and Y are complex-valued, the definitions of autocorrelation and cross correlation are modified to use the conjugate of the first term:

$$R_X(\tau) = E(\overline{X(t)} \cdot X(t+\tau))$$

$$R_{XY}(\tau) = E(\overline{X(t)} \cdot Y(t+\tau))$$

The coherence between X and Y is a normalized function of the density functions:

$$\gamma_{XY}^2(f) = \frac{|S_{XY}(f)|^2}{S_X(f) \cdot S_Y(f)}$$

The partial coherence of X and Y after the effects of a third time series Z have been removed is found using this formidable-looking expression:

$$\gamma_{XY \cdot Z}^2(f) = \frac{\left| S_{XY}(f) - \dfrac{S_{XZ}(f) \cdot S_{YZ}(f)}{S_Z(f)} \right|^2}{\left[S_X(f) - \dfrac{|S_{XZ}(f)|^2}{S_Z(f)} \right] \left[S_Y(f) - \dfrac{|S_{YZ}(f)|^2}{S_Z(f)} \right]}$$

II. Discrete Implementation

Suppose we have N observed values of two processes X and Y. We thus have X(t) and Y(t) defined only for t=0,1,2,...N-1. We then define the correlation functions in terms of an average rather than an expectation:

$$R_X(v) = \frac{1}{N}\sum_{t=0}^{N-v-1} \overline{X(t)}\,X(t+v) \qquad R_{XY}(v) = \frac{1}{N}\sum_{t=0}^{N-v-1} \overline{X(t)}\,Y(t+v)$$

For most purposes it is necessary to determine correlations only for values of v less than or equal to a certain limit L. (Usually L is less than 10% or 20% of the number of observed data points N.) For purposes of computation, extend the sequences X and Y to length N' (which must be greater than N + L) by appending zeros to the end; call these new sequences X' and Y'. Thus,

X'(t)=X(t) and Y'(t)=Y(t) for t=0,1,...N-1

but X'(t)= 0 and Y'(t)= 0 for t=N,N+1,...N'-1

If the fast Fourier transform (FFT) is to be used, N' must be a power of two.

Let $\hat{X}'(u)$ and $\hat{Y}'(u)$ be the Fourier transforms of X' and Y':

$$\hat{X}'(u) = \sum_{t=0}^{N'-1} X'(t)\,e^{2\pi i t u/N'}$$

$$\hat{Y}'(u) = \sum_{s=0}^{N'-1} Y'(s)\,e^{2\pi i s u/N'}$$

Now form a new sequence in which each term is the product of the complex conjugate of the corresponding term of $\hat{X}'$ and the corresponding term of $\hat{Y}'$; then take the inverse Fourier transform of this sequence.

$$C(v) = \frac{1}{N'} \sum_{u=0}^{N'-1} \overline{\{\hat{X}'(u)\}}\, \hat{Y}'(u)\, e^{-2\pi i u v/N'}$$

When we insert the above expressions for X'(u) and Y'(u), and manipulate the (finite) summations, we obtain the following:

$$C(v) = \frac{1}{N'} \sum_{u=0}^{N'-1} \left[\sum_{t=0}^{N'-1} \overline{X'(t)}\, e^{-2\pi i t u/N'} \right] \left[\sum_{s=0}^{N'-1} Y'(s)\, e^{2\pi i s u/N'} \right] e^{-2\pi i u v/N'}$$

$$= \frac{1}{N'} \sum_{u=0}^{N'-1} \sum_{t=0}^{N'-1} \sum_{s=0}^{N'-1} \overline{X'(t)}\, Y'(s)\, e^{2\pi i u(s-t-v)/N'}$$

$$C(v) = \sum_{t=0}^{N'-1} \sum_{s=0}^{N'-1} \overline{X'(t)}\, Y'(s) \left[\frac{1}{N'} \sum_{u=0}^{N'-1} e^{2\pi i u(s-t-v)/N'} \right]$$

It is easily verified that the trigonometric expression in the brackets is equal to zero unless (s-t-v) is zero or an integral multiple (positive or negative) of N', in which case it is equal to one. Because s and t are restricted to the range 0 to N'-1, only two sets of (s,t) pairs meet this criterion.

If s-t-v= -N', then t=N'-v+s, and t must be greater than N'-v. We are restricting v to be less than L, and N' exceeds N+L, so N'-v will be greater than N. However, X'(t)=0 if t is greater than or equal to N, because X' is only an extension of X beyond N-1. Thus, this set of (s,t) pairs contributes nothing to the sum.

If s-t-v=0, then s=t+v and t is restricted to the range 0 to N'-v-1. Our expression then reduces to

$$C(v) = \sum_{t=0}^{N'-v-1} \overline{X'(t)}\, Y'(t+v)$$

Again, because of the way in which the original sequences were extended, Y'(t+v) is nonzero only if t+v is less than N, only if

t is less than or equal to N-v-1. Therefore, N-v-1 may be taken as the upper limit of summation and we have the following relation:

$$C(v) = \sum_{t=0}^{N-v-1} \overline{X(t)}\, Y(t+v) = N \cdot R_{XY}(v)$$

Thus, by performing only three Fourier transforms we can obtain all values of the correlation function which interest us simultaneously. One more transform produces the cross spectral density function. Observe that by substituting X and X' for Y abd Y' the auto-correlation and power spectral density functions of X would have been obtained. By using the fast Fourier transform to perform the above calculations, estimates of spectra may be very efficiently generated on a digital computer. This method was realized in the subroutine CROSS, which can produce either cross correlations or autocorrelations.

III. Smoothing

When a correlation function R(v) (in this section R can be either an autocorrelation or a cross correlation) is Fourier transformed to produce a spectral density function, the explicit relation between R and S is

$$\hat{S}(f) = h\left[R(0) + 2\left(\sum_{k=1}^{L-1} R(k)\cos\left(\frac{\pi k f}{f_c}\right)\right) + R(L)\cos\left(\frac{\pi L f}{f_c}\right)\right]$$

where h is the sampling interval, and f_c is the Nyquist frequency, 1/2h, which is the highest frequency which can be unambiguously determined with a given sampling rate. The $\hat{S}$ indicates that this is a raw estimate of the density.

Unfortunately, the above expression is not a consistent estimate in the sense of mean square convergence. Its variance does not go to zero as the number of sample points increases, and a graph of raw spectral estimates will oscillate wildly about the true values of spectral density.

To improve the spectral estimates, it is necessary to first "smooth" or average the spectral values. A very simple but effective method is to replace each value of the spectral estimate with a weighted average of the original and neighboring values:

$$\tilde{S}(0)=0.5\,\hat{S}(0) + 0.5\,\hat{S}(1)$$

$$\tilde{S}(k)=0.25\,\hat{S}(k-1) + 0.5\,\hat{S}(k) + 0.25\,\hat{S}(k+1) \text{ for } k=1,2,\ldots L-1$$

$$\tilde{S}(L)=0.5\,\hat{S}(L-1) + 0.5\,\hat{S}(L)$$

This smoothing method can be implemented by applying a "window" to the correlation function:

$$R'(v)=R(v)\cdot D(v/L)$$

where D is a weighing function defined as

$$D(u) = \frac{1}{2}(1 + \cos \pi u), \quad |u| \leq 1$$

This is known as a Tukey window. Another possible window, which results in a different degree of smoothing, is the modified Tukey window

$$D(u) = 0.54 + 0.46 \cos \pi u.$$

CHAPTER 2

We consider random processes $\{X_j(t)\}$, $1 \le j \le n$, $-\infty < t < \infty$, that is, for each t, $X_j(t)$ is a random variable, where all $X_j(t)$ are defined on the same sample space. We assume $E(X_i(s)X_j(t+s)) = R_{x_i x_j}(t)$ does not depend on s, where i can equal j. Then

$$R_{x_i x_j}(t) = \int_{-\infty}^{\infty} e^{i2\pi tf} S_{x_i x_j}(f)\, df$$

If $i=j$, $R_{x_i x_i}(t)$ is called autocorrelation function of X_i, and $S_{x_i x_i}(f)$ is the power spectral density of X_i. If $i \ne j$, $R_{x_i x_j}(t)$ is called the cross-correlation function of X_i and X_j, and $S_{x_i x_j}(f)$ is the cross-spectral density function.

Assume $E X_j(t) = 0$ for all j and t.

Also, $S_{x_i x_j}(f) = \int_{-\infty}^{\infty} e^{-i2\pi ft} R_{x_i x_j}(t)\, dt$

There exists a random spectral representation of the $X_j(t)$:

$$X_j(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda}\, dZ_{x_j}(\lambda)$$

where

$$Z_{x_j}(\lambda_2) - Z_{x_j}(\lambda_1) = \underset{T\to\infty}{\text{l.i.m.}} \int_{-T}^{T} \frac{e^{-i2\pi\lambda_2 t} - e^{-i2\pi\lambda_1 t}}{-i2\pi t} X_j(t)\, dt$$

The $Z_{x_j}(\lambda)$ are processes with orthogonal increments and

$$E\left\{\int_{-\infty}^{\infty} f(\lambda)\,dZ_{x_i}(\lambda)\cdot\overline{\int_{-\infty}^{\infty} g(\lambda)\,dZ_{x_j}(\lambda)}\right\} =$$

$$= \int_{-\infty}^{\infty} f(\lambda)\overline{g(\lambda)}\,S_{x_i x_j}(\lambda)\,d\lambda$$

where i can equal j.

$Z_{x_j}(\lambda)$ forms a random spectral measure. Let us consider the physical significance of $S_{xx}(f)$.

Consider a linear time invariant filter with input a stationary process $X(t)$ and output $Y(t) =$

$$\int_{-\infty}^{\infty} e^{i2\pi t\lambda} H(\lambda)\,dZ_x(\lambda).$$

$H(\lambda)$ is called the transfer function. Then

$$R_{YY}(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda}\,|H(\lambda)|^2\,S_{xx}(\lambda)\,d\lambda$$

and

$$S_{YY}(f) = |H(\lambda)|^2\,S_{xx}(f).$$

Also,

$$S_{XY}(f)/S_{xx}(f) = H(f)$$

Hence if we can get estimates $\hat{S}_{XY}(f)$ for $S_{XY}(f)$ and $\hat{S}_{xx}(f)$ of $S_{xx}(f)$, we can get an estimate $\hat{H}(f)$ of $H(f)$:

$$\hat{H}(f) = \frac{\hat{S}_{XY}(f)}{\hat{S}_{xx}(f)}.$$ Also, $$|\hat{H}(f)|^2 = \frac{\hat{S}_{YY}(f)}{\hat{S}_{xx}(f)}$$

is an estimate of the square of the gain $|H(f)|^2$.

Most often we do not have explicit expressions for $H(f)$.

Take for example a system such as a ship which acts like a black box. The waves $X(t)$ act as an input forcing function. The ship processes $X(t)$ in some way and responds by pitching as an output $Y(t)$.

(Sometimes we can write

$$Y(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda} H(\lambda)\, dZ_x(\lambda) =$$

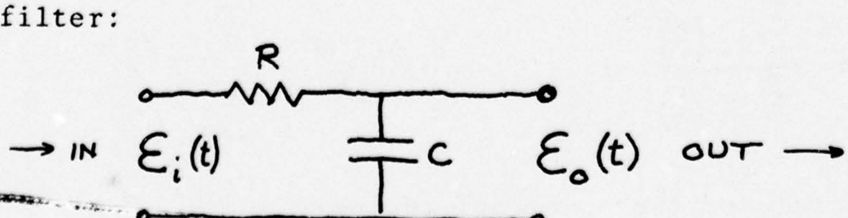
$$= \int_{-\infty}^{\infty} h(\lambda-\mu)\, X(\mu)\, d\mu$$

where

$$H(t) = \int_{-\infty}^{\infty} h(\mu)\, e^{-i2\pi t\mu}\, d\mu .$$

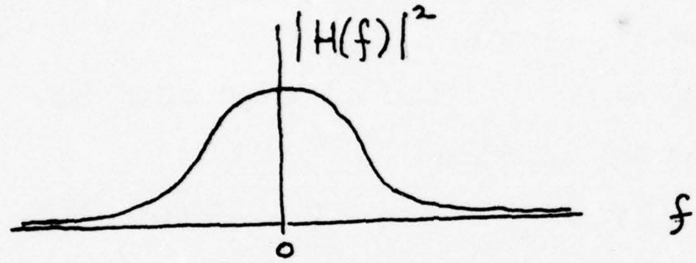
$h(\mu)$ is called impulse response).

We subject the ship model to waves whose spectral distribution ranges over the frequencies it will actually encounter and see how the ship pitches. If the ship encounters waves of its natural pitching frequency, the ship will pitch badly. In this case the design must be adjusted to bring the natural pitching frequency to some frequency at which the waves normally encountered have little energy. The captain could also be warned of the sea conditions under which he will have to alter course or speed to avoid danger-our resonant pitching. As another example consider an RC filter:



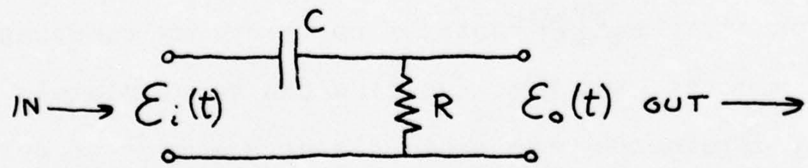
$$\mathcal{E}_o(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda}\left(\frac{1}{1+i\,2\pi RC\lambda}\right) dZ_{\mathcal{E}_i}(\lambda)$$

$$S_{\mathcal{E}_o\mathcal{E}_o}(f) = \left(\frac{1}{1+4\pi^2R^2C^2f^2}\right) S_{\mathcal{E}_i\mathcal{E}_i}(f)$$



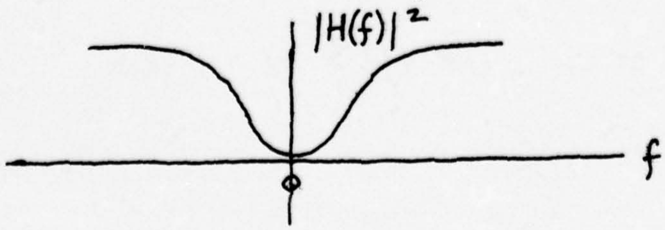
This is a low pass filter. It passes low frequencies and attenuates high frequencies.

For the filter



$$\mathcal{E}_o(t) = \int_{-\infty}^{\infty} e^{i2\pi t\lambda}\left(\frac{i\,2\pi RC\lambda}{1+i\,2\pi RC\lambda}\right) dZ_{\mathcal{E}_i}(\lambda)$$

$$|H(f)|^2 = \frac{4\pi^2R^2C^2f^2}{1+4\pi^2R^2C^2f^2}$$



This is a high pass filter. It passes high frequencies, but attenuates low frequencies. One might wonder how we can estimate spectra for continuous parameter or analog signals by sampling at discrete time points and by using the digital computer recover the spectra. In most applications $S_{xx}(f) \cong 0$ for $|f| \geq W$ for some W.

For high-quality speech

$$S_{xx}(f) \neq 0 \quad , \quad 100 < f < 10{,}000 \ Hz$$

For multichannel telephony

$$S_{xx}(f) \neq 0 \quad , \quad 300 < f < 3400 \ Hz$$

For high-quality music

$$S_{xx}(f) \neq 0 \quad , \quad 30 < f < (10-15) \ KHz.$$

For EEG

$$S_{xx}(f) \cong 0 \ , \quad |f| > 50 \ Hz.$$

The following Sampling Theorum holds:

If a function of time $X(t)$ contains no frequency components higher than W hertz, the time function can be completely specified by determining the ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart. Reconstitution of the original time function, i.e., the signal wave form is possible if the sample pulses are passed through a suitable low pass filter. This is important in time division multiplex systems.

If $S_{xx}(f) \cong 0$ for $|f| > W$, then

$$X(t) = \sum_{n=-\infty}^{\infty} X\left(\frac{n}{2W}\right) \frac{\sin 2\pi W\left(t - \frac{n}{2W}\right)}{2\pi W\left(t - \frac{n}{2W}\right)}$$

Let the discrete time series $X_i(t)$, $t = 0, \pm 1, \pm 2, \ldots$ be obtained by sampling a continuous parameter time series $Y_i(\cdot)$ at time intervals of length h: $X_i(t) = Y_i(t \cdot h)$

Similarly define $X_j(t)$ for $t = 0, \pm 1, \ldots$.

Let

$$R_{x_i x_j}(t) = R_{Y_i Y_j}(t \cdot h)$$

This sampling can be obtained by using a PDP8-E mini-computer, which incorporates an analog-to-digital converter.

Then

$$R_{x_i x_j}(t) = \int_{-\frac{1}{2h}}^{\frac{1}{2h}} S_{x_i x_j}(f) e^{i 2\pi f t h} df$$

where

$$S_{x_i x_j}(f) = \sum_{\ell=-\infty}^{\infty} S_{Y_i Y_j}\left(f + \frac{\ell}{h}\right), \quad -\frac{1}{2h} \le f \le \frac{1}{2h}$$

Thus in order that $S_{x_i x_j}(f) = S_{Y_i Y_j}(f)$

we must have $S_{Y_i Y_j}(f) = 0$ for $|f| > \frac{1}{2h}$.

Otherwise we would get aliasing and frequencies above will be folded back. (One observes this in old westerns where the wheels rotate backwards when the stagecoach starts and slows down). Thus the Nyquist folding frequency

$f_c = W = \frac{1}{2h}$.

For EEG we sample at 10 msec intervals, or to get a power of 2, we sample at $\frac{1}{64}$ or $\frac{1}{128}$ sec. intervals for a period of 10 sec. or less.

Let $X(1), X(2), \ldots X(N)$ be samples taken at time intervals of length h. Assume detrending has been performed. We shall discuss the estimation of $S_{x_i x_j}(f)$.

The sample cross-covariance $\hat{R}_{x_i x_j}(v)$ of lag v between $X_i(\cdot)$ and $X_j(\cdot)$ is defined to be

$$\hat{R}_{x_i x_j}(v) = \frac{1}{N}\sum_{t=1}^{N-v} X_i(t)X_j(t+v) \quad , \quad v = 0,1,2,\ldots,(N-1)$$

$$\hat{R}_{x_i x_j}(v) = \frac{1}{N}\sum_{t=-v+1}^{N} X_i(t)X_j(t+v) \quad , \quad v = -1,-2,\ldots,-(N-1)$$

$$\hat{R}_{x_i x_j}(v) = 0 \quad , \quad |v| \geq N$$

$$\hat{R}_{x_i x_j}(v) = \hat{R}_{x_j x_i}(-v)$$

The $\hat{R}_{x_i x_j}(v)$ are computed by using the FFT.

The sample cross-spectral density function or cross periodogram between $X_i(\cdot)$ and $X_j(\cdot)$ is given by

$$I_{x_i x_j}(f) = \frac{h}{N}\left(\sum_{s=1}^{N} X_i(s)e^{i2\pi f s h}\right)\left(\sum_{t=1}^{N} X_j(t)e^{-i2\pi f t h}\right) =$$

$$= h\sum_{k=-(N-1)}^{N-1} \hat{R}_{x_i x_j}(k)e^{-i2\pi f k h} \quad , \quad -\frac{1}{2h} \leq f \leq \frac{1}{2h} .$$

$$S_{x_i x_j}(f) = h\sum_{\ell=-\infty}^{\infty} R_{x_i x_j}(\ell)e^{-i2\pi \ell f h} .$$

$$E\left(I_{x_i x_i}(f)\right) = E\left(\frac{h}{N}\left|\sum_{t=1}^{N} X_i(t)\, e^{-i2\pi f t h}\right|^2\right) =$$

$$= \frac{h}{N}\int_{-\frac{1}{2h}}^{\frac{1}{2h}} \frac{\sin^2(N\pi(f-\lambda)h)}{\sin^2(\pi(f-\lambda)h)}\, S_{x_i x_i}(\lambda)\, d\lambda \rightarrow$$

$$\rightarrow S_{x_i x_i}(f) \quad , \text{ as } N \rightarrow \infty .$$

But if for example $X_i(\cdot)$ is Gaussian we have approximately

$$P\left[I_{x_i x_i}(f) > x\right] \cong e^{-\frac{1}{S_{x_i x_i}(f)} x}$$

or

$$\frac{2\, I_{x_i x_i}(f)}{S_{x_i x_i}(f)} \sim \chi_2^2 \ ,$$

$$\mathrm{VAR}\left(I_{x_i x_i}(f)\right) \approx S^2_{x_i x_i}(f)$$

The periodogram itself is not a good estimate of the spectrum. It is not a consistent estimate in the sense of mean square convergence. Its variance does not go to zero as N increases. To get a good estimate of $S_{x_i x_i}(f)$ we must "smooth" the periodogram.

Take for the moment $h = \frac{1}{2\pi}$

$$\hat{S}_{x_i x_i}(\lambda) = \int_{-\pi}^{\pi} W(\lambda-\alpha)\, I_{x_i x_i}(\alpha)\, d\alpha$$

$$E(\hat{S}_{x_ix_i}(\lambda)) \longrightarrow \int_{-\pi}^{\pi} W(\lambda-\alpha) S_{x_ix_i}(\alpha)\, d\alpha \cong S_{x_ix_i}(\lambda)$$

$$\text{if} \quad \int_{-\pi}^{\pi} W(\alpha)\, d\alpha = 1$$

$$N \cdot VAR(\hat{S}_{x_ix_i}(\lambda)) = 2\pi \int_{-\pi}^{\pi} W^2(\lambda-\alpha) S^2_{x_ix_i}(\alpha)\, d\alpha + o(1)$$

$$VAR(\hat{S}_{x_ix_i}(\lambda)) \cong \frac{2\pi}{N} S^2_{x_ix_i}(\lambda) \left(\int_{-\pi}^{\pi} W^2(\alpha)\, d\alpha \right)$$

Let $M < N$. Choose $M/N \cong 0.1$ or 0.2

We use estimates

$$\hat{S}_{x_ix_j}(f) = h \sum_{|v| \le M} K\left(\frac{v}{M}\right) \hat{R}_{x_ix_j}(v)\, e^{-i2\pi f v h}$$

$$K(-u) = K(u) \; ; \quad K(0) = 1 , \quad K(u) = 0 \text{ for } |u| > 1$$

$$VAR(\hat{S}_{x_ix_i}(f)) \cong \frac{M}{N} \cdot I \cdot S_{x_ix_i}(f)$$

$$I = \int_{-1}^{1} K^2(u)\, du$$

Approximate confidence intervals for the $S_{x_ix_i}(f)$ can be found by using the fact that the random variable

$$\frac{\nu\, \hat{S}_{x_ix_i}(f)}{S_{x_ix_i}(f)} \cong \chi^2_\nu \qquad \text{where} \quad \nu = \frac{2N}{IM}$$

We choose points $f_K = \dfrac{K f_c}{M}$ for $K = 0, 1, 2, \ldots, M$ to estimate the spectra.

$$f_c = \frac{1}{2h}$$

Then

$$\hat{S}_{x_ix_j}\left(\frac{K}{M} f_c\right) = h \sum_{|v| \le M} K\left(\frac{v}{M}\right) \hat{R}_{x_ix_j}(v)\, e^{-i \frac{2\pi}{2M} v K}$$

The computations for $\hat{R}_{x_i x_j}(v)$ and the sums in $\hat{S}_{x_i x_j}(\frac{K}{M} f_c)$ are performed by efficient use of the FFT. We used the lag window

$$K(u) = \begin{cases} 1 - 6u^2 + 6|u|^3 & , \quad |u| \le \frac{1}{2} \\ 2(1-|u|)^3 & , \quad \frac{1}{2} \le u \le 1 \\ 0 & , \quad u > 1 \end{cases}$$

Then $I = 0.539$, $\nu = \frac{2}{I}\frac{N}{M} = 3.71\frac{N}{M}$

The program MULSPECT estimates the spectra by using the above lag window.

Another method I used involves the use of a modified periodogram and cross-periodograms using cosine taper. Then the modified periodograms are averaged over neighboring points.

A modified periodogram is of the form

$$I^*(f) = \frac{h}{NU}\left|\sum_{j=1}^{N} W_N(j)\,X(j)\,e^{-i2\pi jfh}\right|^2$$

$$U = \frac{1}{N}\sum_{j=1}^{N} W_N^2(j)$$

We have written programs which involve new methods of spectral estimation, namely the fitting of autoregressive schemes to given time series.

The program AUTOREG estimates spectra by fitting autoregressive schemes to time series. We solve for $a_1, a_2, \ldots, a_m$ (with $a_0 = 1$) the following system of equations

$$\sum_{s=0}^{m} a_s \hat{R}_{xx}(t-s) = 0 \quad , \quad t = 1, 2, ---, m$$

$$\hat{R}_{xx}(-t) = \hat{R}_{xx}(t) = \frac{1}{N} \sum_{j=1}^{N-t} X(j) X(j+t)$$

The $\hat{R}_{XX}(t)$ are computed by using subroutine CROSS.

The $a_1, a_2, \ldots, a_m$ are computed by using subroutine LEVNSN.

Let $\hat{\sigma}^2 = \sum_{\kappa=0}^{m} a_\kappa \hat{R}_{xx}(-\kappa)$

The spectral estimate is given by

$$\hat{S}_{xx}(f) = h\hat{\sigma}^2 \frac{1}{\left| \sum_{\kappa=0}^{m} a_\kappa e^{-i2\pi\kappa f h} \right|^2}$$

Pick $B = 2^L \geq m$

Evaluate $\hat{S}_{xx}(f)$ at

$$f = \frac{j}{B} f_c = \frac{j}{B} \frac{1}{2h} \qquad \text{for} \quad j = 0, 1, 2, \ldots, B$$

Let

$$a_{m+1} = a_{m+2} = \cdots = a_{2B-1} = 0$$

$$\hat{S}_{xx}\left(\frac{j}{B} f_c\right) = h\hat{\sigma}^2 \frac{1}{\left| \sum_{\kappa=0}^{2B-1} a_\kappa e^{-i\frac{2\pi}{2B}\kappa j} \right|^2}$$

The program SPCTCLTK estimates spectra by averaging modified periodograms.

The program MAUTOREG estimates multichannel spectra and cross-spectra by fitting multidimensional autoregressive schemes to the multichannel time series.

Let

$$\hat{R}(v) = \begin{bmatrix} \hat{R}_{11}(v) & \cdots & \hat{R}_{1n}(v) \\ \vdots & & \vdots \\ \hat{R}_{n1}(v) & \cdots & \hat{R}_{nn}(v) \end{bmatrix}$$

$$v = 0, 1, 2, \ldots, N-1$$

$$\hat{R}(-v) = \hat{R}^T(v)$$

$$\hat{R}_{jk}(v) = \frac{1}{N} \sum_{t=1}^{N-v} X_j(t) X_k(t+v)$$

$$\hat{S}_x(f) = \begin{bmatrix} \hat{S}_{11}(f) & \cdots & \hat{S}_{1n}(f) \\ \vdots & & \vdots \\ \hat{S}_{n1}(f) & \cdots & \hat{S}_{nn}(f) \end{bmatrix}$$

where $\hat{S}_{jk}(f)$ is an estimate of $S_{x_j x_k}(f)$.

Let $\hat{A}(0) = I_n = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$, $n \times n$ identity matrix.

Let $\hat{A}(j)$ for $j = 1, 2, .., m$ be $n \times n$ matrices which are solutions of the system of matrix equations

$$\sum_{j=0}^{m} \hat{A}(j)\, \hat{R}(j-k) = O$$

for $k = 1, 2, .., m$

(O = $n \times n$ zero matrix)

Let $\hat{V}_m = \sum_{j=0}^{m} \hat{A}(j)\, \hat{R}(j)$

Then we estimate the spectral density matrix by

$$\hat{S}_x(f) = h\left[\sum_{j=0}^{m} \hat{A}(j) e^{i2\pi jfh}\right]^{-1} \hat{V}_m \left(\left[\sum_{j=0}^{m} \hat{A}(j) e^{-i2\pi jfh}\right]^{-1}\right)^T$$

Let $B = 2^L$, $f_c = \frac{1}{2h}$

We evaluate $\hat{S}_x(f)$ at points $f = \frac{k}{B} f_c$

for $k = 0, 1, .., B$

Let $\hat{A}(j) = 0$ for $j = m+1, m+2, ..., 2B-1$

Then

$$\hat{S}_x\left(\frac{k}{B} f_c\right) = h\left[\sum_{j=0}^{2B-1} \hat{A}(j) e^{i \frac{2\pi}{2B} jk}\right]^{-1} \hat{V}_m \left(\left[\sum_{j=0}^{2B-1} \hat{A}(j) e^{-i \frac{2\pi}{2B} jk}\right]^{-1}\right)^T$$

All calculations are performed using FFT.

The elements of the matrix $\hat{R}(v)$ are computed by subroutine MAC, outputted in multiplexed form.
The matrices $\hat{A}(1), \hat{A}(2), \ldots, \hat{A}(m)$ are computed by the subroutine MULLEV.

Several of the programs involve simulation of time series with specified spectral densities. We can simulate EEG or any time series with almost any spectra. Let $\{X(n)\}$ be a sequence of independent observations from N(0,1) (white noise).

Let $Y(t) = \sum_n a_n X(t+n)$ (digital filter).

Then

$$S_{YY}(f) = \left| \sum_n a_n e^{i 2\pi n f h} \right|^2 S_{xx}(f)$$

$$S_{xx}(f) = h \quad \text{for} \quad -\frac{1}{2h} \le f \le \frac{1}{2h}$$

$$S_{YY}(f) = \left| \sum_n a_n e^{i 2\pi n f h} \right|^2 h$$

If we take $\sum_{n=-L}^{L} a_n e^{i 2\pi n f h}$ as the L-th partial sum of the Fourier series of the function

$$\sqrt{\frac{1}{h} S_{YY}(f)}$$

where $S_{YY}(f)$ is a given function, $Y(\cdot)$ will have spectral density $S_{YY}(f)$

Each $X(n)$ can be obtained by taking

$$X(n) = \left(\sum_{j=1}^{12} R_j \right) - 6$$

where $R_1, R_2, \ldots, R_{12}$ are random numbers from the computer.

We took $h = \frac{1}{64}$

Simulation was useful for testing programs.

We define coherence for time series X and Y

$$\gamma^2_{XY}(f) = \frac{|S_{XY}(f)|^2}{S_{XX}(f)\cdot S_{YY}(f)}$$

Partial coherence of $X(\cdot)$ and $Y(\cdot)$ when the effects of $Z(\cdot)$ are removed

$$\gamma^2_{XY.Z}(f) = \frac{\left| S_{XY}(f) - \frac{S_{XZ}(f)\cdot S_{ZY}(f)}{S_{ZZ}(f)} \right|^2}{\left[S_{XX}(f) - \frac{|S_{XZ}(f)|^2}{S_{ZZ}(f)}\right]\left[S_{YY}(f) - \frac{|S_{YZ}(f)|^2}{S_{ZZ}(f)}\right]}$$

Let $\nu = 2n$ be the effective degrees of freedom.

Let $\hat{\gamma}^2(f)$ be an estimate of $\gamma^2(f)$

Let $\hat{\gamma}(f) = +\sqrt{\hat{\gamma}^2(f)}$, $\gamma(f) = +\sqrt{\gamma^2(f)}$

Let $\tanh^{-1}(x) = \frac{1}{2}\ln\frac{1+x}{1-x}$, $|x| < 1$

Then $\tanh^{-1}(\hat{\gamma}(f))$, where $\hat{\gamma}(f) = \hat{\gamma}_{XY}(f)$,

has approximately a normal distribution:

$$\tanh^{-1}(\hat{\gamma}(f)) \sim N\left(\tanh^{-1}(\gamma(f)) + \frac{1}{2(n-1)}, \frac{1}{2(n-1)}\right)$$

provided $n > 20$, $0.4 \le \gamma^2(f) \le 0.95$

If $f \ne 0, \frac{1}{2h}$, and if $\gamma(f) = 0$,

then

$$(n-1)\frac{\hat{\gamma}^2(f)}{(1-\hat{\gamma}^2(f))} = F_{2,2(n-1)}$$

where F_{m_1, m_2} is a random variable having an F distribution with degrees of freedom m_1 and m_2. This can be used to test the null hypothesis $H_0 : \gamma(f) = 0$ against the alternative $H_1 : \gamma(f) > 0$.
If $f \neq 0$, $\hat{\gamma}_{XY \cdot Z}(f)$ has the same distribution as $\hat{\gamma}_{XY}(f)$.

Spectra and cross-spectra and partial coherences can be used to localize brain tumors and epileptogenic foci in the brain.

Suppose we want to test whether Z drives X and Y. (Z might be an epileptogenic focus).
Assume

$\gamma^2_{XY}(f)$, $\gamma^2_{XZ}(f)$, $\gamma^2_{YZ}(f)$,

$S_{XX}(f)$, $S_{YY}(f)$, $S_{ZZ}(f)$ are significantly different from zero over a certain frequency range.
Assume $\gamma^2_{XZ \cdot Y}(f) \neq 0$, $\gamma^2_{YZ \cdot X}(f) \neq 0$, but $\gamma^2_{XY \cdot Z}(f) = 0$.

Then we would suspect that Z drives X and Y.

Suppose we want to test whether Z drives $X_1, X_2, \ldots, X_n$. Apply the above analysis to all possible subsets of the recordings taken three at a time. Suppose all the spectra

$S_{X_iX_j}(f)$, $S_{ZZ}(f)$, $\gamma^2_{X_iX_j}(f)$, $\gamma^2_{X_iZ}(f)$, $S_{X_iX_i}(f)$, $\gamma^2_{X_iZ \cdot X_j}(f)$, $i \neq j$,

are nonzero for all i, j, but $\gamma^2_{X_iX_j \cdot Z}(f) = 0$ for $i \neq j$.

Then we would suspect strongly that Z drives $X_1, X_2, \ldots, X_n$. The previously described partial coherence spectral analysis can be extended to a large number of data channels to test whether a linear combination of channels drives other channels.

The multichannel coherence spectra and partial coherence spectra are computed by the program SPCTBGTK. The program also plots the partial coherence spectra as well as the coherence spectra.

Bibliography

Anderson, T. W., The Statistical Analysis of Time Series, John Wiley & Sons, Inc., 1971

Bendat, J. S. and Piersol, A. G., Random Data: Analysis and Measurement Procedures, Wiley-Interscience, 1971

Brigham, E. O., The Fast Fourier Transform, Prentice-Hall, Inc., 1974

Doob, J. L., Stochastic Processes, John Wiley & Sons, Inc., 1953, Chaps: 9, 10, 11

Grenander, U. and Rosenblatt, M., Statistical Analysis of Stationary Time Series, John Wiley & Sons, Inc., 1957

Hannan, E. J., Multiple Time Series, John Wiley & Sons, Inc., 1970

Jenkins, G. M. and Watts, D. G., Spectral Analysis and Its Applications, Holden-Day, 1969

Koopmans, L. H., "On the coefficient of coherence for weakly stationary stochastic processes." Ann. Math. Statist. 35, pp 532-549 (1964)

Koopmans, L. H., "On the multivariate analysis of weakly stationary stochastic processes." Ann. Math. Statist, 35, pp 1765-1780 (1964)

Koopmans, L. H., The Spectral Analysis of Time Series, Academic Press, 1974

Olshen, A. O., "Asymptotic Properties of the Periodogram of a Discrete Stationary Process," J. Appl. Prob. 4, pp 508-528 (1967)

Otnes, R. K. and Enochson, L., Digital Time Series Analysis, John Wiley & Sons, Inc., 1972

Panter, P. F., Modulation, Noise and Spectral Analysis, McGraw-Hill Book Co., 1965

Parzen, E., Time Series Analysis Papers, Holden-Day, 1967

Yaglom, A. M., An Introduction to the Theory of Stationary Random Functions, Prentice-Hall, Inc., 1962

APPENDIX A

PROGRAM LISTINGS

MULSPECT

```
100 * DIMENSION X(NS*LX),R1(LR*NS*NS),W(M),F(M),S(M*NS*NS)
110 * DIMENSION C(M1*NS*NS),TLAG(2*LR-1),Z(2*LR-1),TLAGA(LR)
120 * RS=NUMBER OF CHANNELS
130 * LX=NUMBER OF DATA POINTS FROM EACH CHANNEL
140 * M1=M+1=LENGTH OF TIME LAG
150 * LR=MAXIMUN DESIRED TIME LAG .LE. LX
160 * L=SMALLEST INTEGER SUCH THAT LX <=2**L
170 * M=MAXIMUM LAG, M=2**(N-1)
180 * N IS DEFINED SO THAT 2*M=2**N
190 * H=LENGTH OF SAMPLING INTERVAL
200 *  LNXS=LX*NS
210 * LRNSNS=LR*NS*NS
220 * M1NSNS=M1*NS*NS
230 DIMENSION X(320,2),R1(50,2,2),W(32),F(33),S(142),C(33,2,2),TLAG(99)
240 &,Z(99),TLAGA(50)
250 CHARACTER CH(2)/"1","2"/
260 DATA NS,LX,M,M1,LR,L,N,LXNS,LRNSNS,IN/2,320,32,33,50,9,6,640,200,2/
270 LIBRARY "REMAV","NLOGN","CROSS","WPARZ","MACOR","COQUAD","MOVE"
280 &,"NORMAG","COHERE","OLDPLO","GFSORT","BIG","SMALL","PLOTTR"
290 H=1./64.
300 OPENFILE 2,"NTIDAT","NUMERIC"
310 READ(2)X
315 CALL WPARZ(M,W)
320 DO 1 J=1,NS
330 1 CALL REMAV(LX,X(1,J))
340 CALL MACOR(NS,LX,X,LR,R1,LXNS,LRNSNS,L)
350 CALL COQUAD(H,NS,M,N,W,R1,S,M1,LR)
360 CALL COHERE(M1,NS,S,C)
370 DO 7 J=1,M1
380 7 F(J)=J-1
390 DO 500 J=1,NS-1
400 DO 500 K=J+1,NS
410 WRITE(0,300)CH(J),CH(K)
420 300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
430 CALL OLDPLO(C(1,J,K),F,M)
450 100 FORMAT(1H ,2(F6.2,4X,E9.2,6X)/)
460 WRITE(0,102)
470 102 FORMAT(5(1H  ,/))
480 500 CONTINUE
490 DO 105 J=1,NS
500 WRITE(0,107)CH(J)
510 107 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
520 CALL OLDPLO(C(1,J,J),F,M)
540 105 WRITE(0,102)
550 DO 700 I=1,NS-1
560 DO 700 J=I+1,NS
570 WRITE(0,901)CH(I),CH(J)
580 901 FORMAT(' CROSS CORRELATION BETWEEN CHANNELS ',A1,' AND ',A1)
590 DO 108 L=1,LR-1
600 TLAG(L)=L-LR
```

MULSPECT (continued)

```
108 Z(L)=R1(LR-L+1,J,I)
DO 109 L=LR,LR+LR-1
TLAG(L)=L-LR
109 Z(L)=R1(L-LR+1,I,J)
CALL PLOTTR(Z,TLAG,LR+LR-1)
WRITE(0,102)
700 CONTINUE
DO 701 K=1,LR
701 TLAGA(K)=K-1
DO 801 J=1,NS
WRITE(0,501)CH(J)
501 FORMAT(' AUTO CORRELATION FOR CHANNEL ',A1)
CALL PLOTTR(R1(1,J,J),TLAGA,LR)
801 WRITE(0,102)
PRINT,X
STOP
END
```

SPCTRGTK

```
100 * NS=EFFECTIVE NUMBER OF SCANS READ IN
110 * NV=NUMBER OF CHANNELS
120 * NB=NUMBER OF FREQUENCY BANDS (A POWER OF 2)
140 * NSCANS=N=THE LEAST POWER OF 2 .GE.NS
150 * SR=SAMPLING RATE=1/H
160 * X-INPUT SERIES(ARRAY)
170 * P-ARRAY FOR STORING CROSS SPECTRA
180 * IDIMP=NV1*(NV1+1)*(NB+1)
190 DIMENSION X(1024,4),P(660)
195  REAL F(33),W2(33,4,4,4),SP(33)
200 COMPLEX S(33,4,4),S113,S223,S123,S112,S332,S132,S221,S331,S231
210 CHARACTER CH(4)/"1","2","3","4"/
220   DATA NS,NV,NB,SR,PI/800,4,32,64.0,3.14159265/
225 LIBRARY "CCAB"
230 LIBRARY "FAST","TRANS","OLDPLO","GFSORT","BIG","SMALL"
240  OPENFILE 2,"FCHDAT","NUMERIC"
250 READ(2)((X(J,I),J=1,NS),I=1,NV)
260 * DETREND EACH SERIES BY SUBTRACTING FROM EACH SERIES
270 * ITS LEAST SQUARES LINEAR REGRESSION LINE
280 FNS=NS
290 TBAR=0.5*(FNS+1.)
300 TSUMSQ=(FNS*(FNS+1.)*(FNS+FNS+1.))/6.
310 DO 76 I2=1,NV
320 SUM=0.
330 CRSPRO=0.
340 DO 77 I1=1,NS
350 SUM=SUM+X(I1,I2)
360 77 CRSPRO=CRSPRO+FLOAT(I1)*X(I1,I2)
370 FMEAN=SUM/FNS
380 BETA=(CRSPRO-FNS*TBAR*FMEAN)/(TSUMSQ-FNS*TBAR*TBAR)
390 DO 76 I1=1,NS
400 76 X(I1,I2)=X(I1,I2)-FMEAN-BETA*(I1-TBAR)
430 * WINDOW EACH SERIES WITH A COSINE TAPER
440 IR=NS/10
450 R=IR
460 DO 80 I1=1,IR
470 FI1=I1
480 FINT=FI1-0.5
490 WINDOW=0.5*(1.0-COS(PI*FINT/R))
500 I3=NS+1-I1
510 DO 80 I2=1,NV
520 X(I1,I2)=WINDOW*X(I1,I2)
530 80 X(I3,I2)=WINDOW*X(I3,I2)
550 LOG2NS=0
560 NSCANS=1
570 54 IF(NS.LE.NSCANS)GO TO 55
580 LOG2NS=LOG2NS+1
590 NSCANS=NSCANS+NSCANS
600 GO TO 54
620 55 IF(NS.EQ.NSCANS)GO TO 74
```

SPCTRGTK (continued)

```
* ZERO OUT THE LAST PART OF EACH SERIES IF THE NUMBER OF SCANS
* IS NOT A POWER OF 2
I1BEGN=NS+1
DO 75 I1=I1BEGN,NSCANS
DO 75 I2=1,NV
75 X(I1,I2)=0.
  74 CONTINUE
IF(MOD(NV,2))70,82,70
* IF THE NUMBER OF SERIES IS ODD, FILL A DUMMY SERIES WITH ZEROS
70 NV1=NV+1
DO 83 I1=1,NSCANS
83 X(I1,NV1)=0.0
   GO TO 85
   82 NV1=NV
  85 CONTINUE
    IDIMP=NV1*(NV1+1)*(NB+1)
CALL TRANS(P,IDIMP,X,NSCANS,NV1,NB,LOG2NS)
* CROSS SPECTRA ESTIMATES ARE IN ARRAY P
* THE CROSS SPECTRAL ESTIMATES IN ARRAY P ARE SCALED BY MULTIPLUING
* BY C1
WNDPWR=FNS-1.25*R
FSCANS=NSCANS
FNB=NB
FD=FSCANS/(FNB+FNB)
C1=0.25/(SR*(FD+1.)*WNDPWR)
IROWSP=NB+NB+2
ICOLSP=(NV1*(NV1+1))/2
ISIZEP=IROWSP*ICOLSP
DO 95 I1=1,ISIZEP
95 P(I1)=C1*P(I1)
   NB1=NB+1
DO 200 J=1,NV
IX=2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2-J)
DO 200 K=J,NV
IJK=IX+2*NB1*K
DO 200 I=1,NB1
 200 S(I,J,K)=CMPLX(P(IJK+I+I-1),(-1.0)*P(IJK+I+I))
DO 201 J=1,NV-1
DO 201 K=J+1,NV
DO 201 I=1,NB1
CSS=CCAB(S(I,J,J))*CCAB(S(I,K,K))
   IF(CSS-1.0E-07) 17,18,18
17 S(I,K,J)=(0.0,0.0)
GO TO 201
18 S(I,K,J)=CMPLX(CCAB(S(I,J,K))**2/CSS,0.0)
201 CONTINUE
DO 202 J1=1,NV-2
DO 202 J2=J1+1,NV-1
DO 202 J3=J2+1,NV
DO 202 I=1,NB1
```

SPCTBGTK (continued)

```
RES=REAL(S(I,J3,J3))
  S113=S(I,J1,J1)*(1.0-S(I,J3,J1))
S223=S(I,J2,J2)*(1.-S(I,J3,J2))
IF(RES-1.0E-07) 901,902,902
901 S123=S(I,J1,J2)
GO TO 903
902 S123=S(I,J1,J2)-S(I,J1,J3)*CONJG(S(I,J2,J3))/RES
903 IF(CCAB(S113)*CCAB(S223)-1.0E-07) 601,602,602
601 W2(I,J1,J2,J3)=0.0
GO TO 202
 602 W2(I,J1,J2,J3)=(CCAB(S123)**2)/(CCAB(S113)*CCAB(S223))
RES=REAL(S(I,J2,J2))
S112=S(I,J1,J1)*(1.0-S(I,J2,J1))
S332=S(I,J3,J3)*(1.-S(I,J3,J2))
IF(RES-1.0E-07) 1001,1002,1002
1001 S132=S(I,J1,J3)
GO TO1003
1002 S132=S(I,J1,J3)-S(I,J1,J2)*S(I,J2,J3)/RES
1003 IF(CCAB(S112)*CCAB(S332)-1.0E-07) 701,702,702
701 W2(I,J1,J3,J2)=0.0
GO TO 202
702 W2(I,J1,J3,J2)=(CCAB(S132)**2)/(CCAB(S112)*CCAB(S332))
RES=REAL(S(I,J1,J1))
S221=S(I,J2,J2)*(1.-S(I,J2,J1))
S331=S(I,J3,J3)*(1.-S(I,J3,J1))
IF(RES-1.0E-07) 1101,1102,1102
1101 S231=S(I,J2,J3)
GO TO 1103
1102 S231=S(I,J2,J3)-S(I,J1,J3)*CONJG(S(I,J1,J2))/RES
1103 IF(CCAB(S221)*CCAB(S331)-1.0E-07) 801,802,802
801 W2(I,J2,J3,J1)=0.0
GO TO 202
   802 W2(I,J2,J3,J1)=(CCAB(S231)**2)/(CCAB(S221)*CCAB(S331))
202 CONTINUE
CONTINUE
DO 7 J=1,NB1
7 F(J)=J-1
DO 500 J=1,NV-1
DO 500 K=J+1,NV
WRITE(0,300)CH(J),CH(K)
300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
DO 319 I=1,NB1
  SP(I)=REAL(S(I,K,J))
319 CONTINUE
 CALL OLDPLO(SP,F,NB)
  WRITE(0,102)
102 FORMAT(25(1H ,/))
DO 500 L=1,NV
IF((J-L)*(K-L))418,500,418
 418  DO 512 I=1,NB1
```

SPCTBGTK (continued)

```
 SP(I)=W2(I,J,K,L)
512 CONTINUE
WRITE(0,301) CH(J),CH(K),CH(L)
301 FORMAT(' PARTIAL COHERENCE BETWEEN CHANNELS ',A1,' AND ',
&A1,/,' AFTER THE INFLUENCE OF CHANNEL ',A1,' HAS BEEN REMOVED')
CALL OLDPLO(SP,F,NB)
WRITE(0,102)
500 CONTINUE
DO 417 J=1,NV
WRITE(0,317)CH(J)
317 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
IX=2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2)-1
DO 416 I=1,NB1
 416  SP(I)=P(IX+I+I)
  CALL OLDPLO(SP(1),F,NB)
417 WRITE(0,102)
STOP
END
```

SPCTCLTK

```
100 * NS=EFFECTIVE NUMBER OF CHANNELS READ IN
110 * NV=NUMBER OF CHANNELS
120 * NB=NUMBER OF FREQUENCY BANDS(A POWER OF 2)
140 * JSCANS  IS ALSO A POWER OF 2
150 * SR=SAMPLING RATE=1/H
160 * X=INPUT SERIES(ARRAY)
170 * P=ARRAY FOR STORING CROSS SPECTRA
180 DIMENSION X(1024,2),P(198),S(132),C(132),F(33)
185 DIMENSION S1(33,2,2)
190 CHARACTER CH(2)/"1","2"/
200 EQUIVALENCE (S,C)
201 EQUIVALENCE (S,S1)
210 DATA NS,NV,NB,JSCANS,SR,PI/320,2,32,1024,64.,3.14159265/
215 LIBRARY "FAST","TRANS","MOVE","NORMAG","COHERE","PLOT","GFSORT"
216 &,"BIG","SMALL"
220 OPENFILE 2,"NTIDAT","NUMERIC"
230 READ(2)((X(J,I),J=1,NS),I=1,NV)
240 * DETREND THE SERIES BY SUBTRACTING FROM EACH SERIES ITS
250 * LEAST SQUARES LINEAR REGRESSION LINE
260 FNS=NS
270 TBAR=0.5*(FNS+1.0)
280 TSUMSQ=(FNS*(FNS+1.0)*(FNS+FNS+1.0))/6.0
290 DO 76 I2=1,NV
300 SUM=0.0
310 CRSPRO=0.0
320 DO 77 I1=1,NS
330 SUM=SUM+X(I1,I2)
340 77 CRSPRO =CRSPRO+FLOAT(I1)*X(I1,I2)
350 FMEAN=SUM/FNS
360 BETA=(CRSPRO-FNS*TBAR*FMEAN)/(TSUMSQ-FNS*TBAR*TBAR)
370 DO 78 I1=1,NS
380 FREG=FMEAN+BETA*(FLOAT(I1)-TBAR)
390 78 X(I1,I2)=X(I1,I2)-FREG
400 76 CONTINUE
410 * WINDOW EACH SERIES WITH A COSINE TAPER
420 IR=NS/10
430 R=IR
440 DO 79 I1=1,IR
450 FI1=I1
455 FINT=FI1-0.5
460 WINDOW=0.5*(1.0-COS(PI*FINT/R))
470 I3=NS+1-I1
480 DO 80 I2=1,NV
490 X(I1,I2)=WINDOW*X(I1,I2)
500 80 X(I3,I2)=WINDOW*X(I3,I2)
510 79 CONTINUE
520 LOG2NS=0
530 NSCANS=1
540 54 IF(NS.LE.NSCANS)GO TO 55
550 LOG2NS=LOG2NS+1
```

SPCTCLTK (continued)

```
NSCANS=NSCANS+NSCANS
GO TO 54
55 IF(NS.EQ.NSCANS)GO TO 74
* ZERO OUT THE LAST PART OF EACH SERIES IF THE NUMBER OF
* SCANS IS NOT A POWER OF 2
I1BEGN=NS+1
DO 75 I1=I1BEGN,NSCANS
DO 75 I2=1,NV
75 X(I1,I2)=0.0
74 CONTINUE
IF(MOD(NV,2))70,82,70
* IF THE NUMBER OF SERIES IS ODD, FILL A DUMMU SERIES WITH ZEROS
70 NV1=NV+1
DO 83 I1=1,NSCANS
83 X(I1,NV1)=0.0
GO TO 85
82 NV1=NV
85 CONTINUE
IDIMP=NV1*(NV1+1)*(NB+1)
 CALL TRANS(P,IDIMP,X,NSCANS,NV1,NB,LOG2NS)
* CROSS SPECTRUM ESTIMATES ARE IN ARRAY P
* THE CROSS SPECTRUM ESTIMATES IN ARRAY P ARE SCALED BY
* MULTIPLYING BY C
WNDPWR=FNS-1.25*R
FSCANS=NSCANS
FNB=NB
FD=FSCANS/(FNB+FNB)
C1=0.25/(SR*(FD+1.0)*WNDPWR)
IROWSP=NB+NB+2
ICOLSP=(NV1*(NV1+1))/2
ISIZEP=IROWSP*ICOLSP
DO 95 I1=1,ISIZEP
95 P(I1)=C1*P(I1)
NB1=NB+1
DO 1000 J=1,NV1
DO 1000 K=J,NV1
DO 1000 I=1,NB1
S1(I,J,K)=P(2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2+K-J)+I+I-1)
IF(J-K)99,1000,1000
99 S1(I,K,J)=P(2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2+K-J)+I+I)
1000 CONTINUE
CALL COHERE(NB1,NV1,S,C)
DO 7 J=1,NB1
7 F(J)=J-1
DO 500 J=1,NV-1
DO 500 K=J+1,NV
WRITE(0,300)CH(J),CH(K)
300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
CALL PLOT(C(1+NB1*(J-1)+NB1*NV1*(K-1)),F,NB)
100 FORMAT(1H ,F5.0,4X,E9.2/)
```

SPCTCLTK (continued)

```
WRITE(0,102)
102 FORMAT(5(1H ,/))
500 CONTINUE
DO 105 J=1,NV
WRITE(0,107)CH(J)
107 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
CALL PLOT(C(1+NB1*(J-1)+NB1*NV1*(J-1)),F,NB)
105 CONTINUE
STOP
END
```

AUTOREG

```
100 LIBRARY "NLOGN","CROSS","REMAV","LEVNSN","FFT","CCAB","OLDPLO","GFSORT"
101 &,"BIG","SMALL"
110 * LR=2*NB
120 DIMENSION X(320),R1(64),A(64),G(33),F(33)
130 COMPLEX AC(64)
140 * 2**LL IS THE SMALLEST POWER OF 2 WITH LX.LE.2**LL
150 DATA LX, NB,LR,LL,L1/320,32,64,9,6/
160 DATA H/0.015625/
170 OPENFILE 3,"NPUB","NUMERIC"
180 READ(3)(X(I),XX,I=1,320)
190 CALL REMAV(LX,X)
200 CALL CROSS(LX,X,X,LR,R1,LL)
210 CALL LEVNSN(LR,R1,A,S,M)
220 NB1=NB+1
230 M1=M+1
240 DO 2 L=1,M1
250 2 AC(L)=CMPLX(A(L),0.0)
260 M2=M1+1
265 NB2=NB+NB
270 DO 11 L=M2,NB2
280 11 AC(L)=(0.0,0.0)
290 * 2*NB=2**L1
300 CALL FFT(AC,L1)
310 DO 3 K=1,NB1
320 3 G(K)=(H*S)/CABS(AC(K))**2
330 DO 1 J=1,33
340 1 F(J)=J-1
350 CALL OLDPLO(G,F,NB)
370 100 FORMAT(1H0,F5.2,4X,E9.2/)
380 END
```

MAUTOREG

```
100 LIBRARY "CROSS","MOVE","NORMAG","REMAV","OLDPLO"
101 &,"GFSORT","BIG","SMALL","RZERO","ZERO","COHERE","MAINV"
102 &,"MOVEC","BRAINY","MATMUL","FFT"
103 &,"MULLEV"
110 * LR=2*NB,LXNS=LX*NS
120 * LM=SMALLEST INTEGER SUCH THAT LX.LE.2**LM
125 DIMENSION BBB(2047,2)
130 DIMENSION X(1024,2),R1(2,2,64),A(2,2,64),AP(2,2,64),B(2,2,64)
131 &,BP(2,2,64),VA(2,2),VB(2,2),DA(2,2),DB(2,2),CA(2,2),CB(2,2)
132 &,S(33,2,2),C(33,2,2),F(33)
140 COMPLEX AC(64),S1(2,2,33),CDB(2,2),ST(2,2)
141 &,CVA(2,2)
150 CHARACTER CH(2)/"1","2"/
160 EQUIVALENCE (S,C)
165 EQUIVALENCE (NS,NV)
170 DATA LX,LXNS/1024,2048/
180 DATA NS,NB,LR,L1,LM/2,32,64,6,10/
190 OPENFILE 2,"NPUB","NUMERIC"
200 READ(2) (BBB(J,1),J=1,2047)
210 OPENFILE 3,"NCH2","NUMERIC"
220 READ(3) (BBB(J,2),J=1,2047)
222  DO 1991 I=1,2
224  DO 1991 J=1,1024
226  1991 X(J,I)=BBB(1+2*(J-1),I)
230 DO 7 J=1,NS
240 7 CALL REMAV(LX,X(1,J))
250 CALL MAC(NS,LX,X,LR,R1,LXNS,LM)
260 CALL MULLEV(NS,LR,R1,A,AP,B,BP,VA,VB,DA,DB,CA,CB,M)
270 DO 5 I=1,NS
280 DO 5 J=1,NS
290 5 CVA(I,J)=CMPLX(VA(I,J),0.0)
300 NB1=NB+1
310 DO 1 I=1,NS
320 DO 1 J=1,NS
330 M1=M+1
340 DO 2 L=1,M1
350 2 AC(L)=CMPLX(A(I,J,L),0.0)
360 M2=M1+1
365 NB2=NB+NB
370 DO 11 L=M2,NB2
380 11 AC(L)=(0.0,0.0)
390 * 2*NB=2**L1
400 CALL FFT(AC,L1)
410 DO 1 K=1,NB1
420 1 S1(I,J,K)=AC(K)
430 DO 4 K=1,NB1
440 CALL MAINV(NS,S1(1,1,K),CDB)
450 DO 3 I=1,NS
460 DO 3 J=1,NS
470 3 S1(I,J,K)=CONJG(CDB(J,I))
```

MAUTOREG (continued)

```
480 CALL MATMUL(NS,CDB,CVA,ST)
490 CALL MATMUL(NS,ST,S1(1,1,K),CDB)
500 DO 4 I=1,NS
510 DO 4 J=1,NS
520 4 S1(I,J,K)=CDB(I,J)
530 DO 10 I=1,NS
540 DO 10 J=I,NS
550 DO 10 K=1,NB1
560 S(K,I,J)=REAL(S1(I,J,K))
570 IF(J-I)9,10,10
580 9 S(K,J,I)=-AIMAG(S1(I,J,K))
590 10 CONTINUE
950 CALL COHERE(NB1,NS,S,C)
960 DO 97 J=1,NB1
970 97 F(J)=J-1
980 DO 500 J=1,NV-1
990 DO 500 K=J+1,NV
1000 WRITE(0,300)CH(J),CH(K)
1010 300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
1020 CALL OLDPLO(C(1,J,K),F,NB)
1040 100 FORMAT(1H ,F5.0,4X,E9.2/)
1050 WRITE(0,102)
1060 102 FORMAT(25(1H ,/))
1070 500 CONTINUE
1080 DO 105 J=1,NV
1085 WRITE(0,102)
1090 WRITE(0,107)CH(J)
1100 107 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
1110 CALL OLDPLO(C(1,J,J),F,NB)
1125 105 CONTINUE
1130 STOP
1140 END
1200 SUBROUTINE ESCAL(N,A,B)
1210 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
1220 DIMENSION A(N,N),B(N,N)
1230 DO 5 I=1,N
1240 DO 5 J=1,N
1250 5 B(I,J)=0.0
1260 B(1,1)=1./A(1,1)
1270 IF(N.EQ.1)RETURN
1280 DO 40 M=2,N
1290 K=M-1
1300 EK=A(M,M)
1310 DO 10 I=1,K
1320 DO 10 J=1,K
1330 10 ED=EK-A(M,I)*B(I,J)*A(J,M)
1340 B(M,M)=1./EK
1350 DO 30 I=1,K
1360 DO 20 J=1,K
1370 20 B(I,M)=B(I,M)-B(I,J)*A(J,M)/EK
```

MAUTOREG (continued)

```
1380 30 B(M,I)=B(I,M)
1390 DO 40 I=1,K
1400 DO 40 J=1,K
1410 40 B(I,J)=B(I,J)+B(I,M)*B(M,J)*EK
1420 RETURN
1430 END
1500 SUBROUTINE FADDEJ(N,A,AINV,DET,ADJUG,P)
1510 LIBRARY "CCAB"
1520 DIMENSION A(N,N),AINV(N,N),ADJUG(N,N),P(N)
1530 COMPLEX A,AINV,DET,ADJUG,P
1540 COMPLEX PN
1550 NN=N*N
1560 CALL MOVEC(NN,A,AINV)
1570 DO 4 K=1,N
1580 P(K)=(0.0,0.0)
1590 DO 2 I=1,N
1600 2 P(K)=P(K)+AINV(I,I)
1610 P(K)=P(K)/FLOAT(K)
1620 IF(K.EQ.N)GO TO 5
1630 CALL MOVEC(N*N,AINV,ADJUG)
1640 DO 3 I=1,N
1650 3 ADJUG(I,I)=AINV(I,I)-P(K)
1660 4 CALL BRAINY(N,N,1,A,N,N,1,ADJUG,AINV,1)
1670 5 CALL MOVEC(N*N,ADJUG,AINV)
1680 E30=1.0E-30
1690 IF(CCAB(P(N)).LT.E30)GO TO 7
1700 DO 6 I=1,N
1710 DO 6 J=1,N
1720 PN=P(N)
1730 6 AINV(I,J)=AINV(I,J)/PN
1740 7 DET=P(N)
1750 IF(MOD(N,2).EQ.1)RETURN
1760 DET=-DET
1770 DO 8 I=1,N
1780 DO 8 J=1,N
1790 8 ADJUG(I,J)=-ADJUG(I,J)
1800 RETURN
1810 END
1900 SUBROUTINE ESCALD(N,A,B,DET)
1910 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
1920 * ALSO COMPUTES DETERMINANT OF A
1930 DIMENSION A(N,N),B(N,N)
1940 DO 5 I=1,N
1950 DO 5 J=1,N
1960 5 B(I,J)=0.0
1970 B(1,1)=1./A(1,1)
1980 DET=A(1,1)
1990 IF(N.EQ.1)RETURN
2000 DO 40 M=2,N
2010 K=M-1
```

MAUTOREG (continued)

```
EK=A(M,M)
DO 10 I=1,K
DO 10 J=1,J
10 EK=EK-A(M,I)*B(I,J)*A(J,M)
DET=DET*EK
B(M,M)=1./EK
DO 30 I=1,K
DO 20 J=1,K
20 B(I,M)=B(I,M)-B(I,J)*A(J,M)/EK
30 B(M,I)=B(I,M)
DO 40 I=1,K
DO 40 J=1,K
40 B(I,J)=B(I,J)+B(I,M)*B(M,J)*EK
RETURN
END
SUBROUTINE SIMEQ(M,N,A,B,C)
* NMAX=LARGEST VALUE OF N TO BE PROCESSED
* NONDUMMY DIMENSION S(NMAX,NMAX)
* FOR EXAMPLE, IF NMAX=4 THEN
DIMENSION S(4,4)
DIMENSION A(M,N),B(N,N),C(M,N)
CALL MOVE(N*N,B,S)
CALL ESCAL(N,S,B)
DO 1 I=1,M
DO 1 J=1,N
A(I,J)=0.0
DO 1 K=1,N
1 A(I,J)=A(I,J)+C(I,K)*B(K,J)
CALL MOVE(N*N,S,B)
RETURN
END
SUBROUTINE SIMEQD(M,N,A,B,C,D)
* NMAX=LARGEST VALUE OF N TO BE PROCESSED
* NONDUMMY DIMENSION A(NMAX,NMAX)
* FOR EXAMPLE, IF NMAX=4, THEN
DIMENSION S(4,4)
DIMENSION A(M,N),B(N,N),C(M,N)
CALL MOVE(N*N,B,S)
CALL ESCALD(N,S,B,D)
DO 1 I=1,M
DO 1 J=1,N
A(I,J)=0.0
DO 1 K=1,N
1 A(I,J)=A(I,J)+C(I,J)*B(K,J)
CALL MOVE(N*N,S,B)
RETURN
END
SUBROUTINE MAC(NS,LX,X,LR,R1,LXNS,L)
* SUBROUTINE MAC COMPUTES THE MULTICHANNEL AUTOCORRELATION OF
* THE NS CHANNEL TIME SERIES X
```

MAUTOREG (continued)

```
2630 * LX=LENGTH OF EACH TIME SERIES IN EACH CHANNEL
2640 * NS=NUMBER OF CHANNELS
2650 * LR=DESIRED LENGTH OF CORRELATION, LR.LE.2**L
2660 * L IS SMALLEST INTEGER SUCH THAT LX.LE.2**L
2670 * LXNS=LX*NS
2680 * NONDUMMY DIMENSION S(LF), WHERE LF.GE.ANTICIPATED LR
2690 DIMENSION X(LXNS),R1(NS,NS,LR)
2700 DIMENSION S(64)
2710 DO 1 I=1,NS
2720 I1=1+(I-1)*LX
2730 DO 1 J=1,NS
2740 J1=1+(J-1)*LX
2750 CALL CROSS(LX,X(I1),X(J1),LR,S,L)
2760 DO 1 K=1,LR
2770 1 R1(J,I,K)=S(K)
2780 RETURN
2790 END
```

BIG

```
100 FUNCTION BIG(A,M)
110 DIMENSION A(M)
120 B=A(1)
130 DO 1 K=2,M
140 IF(A(K)-B)1,1,2
150 2 B=A(K)
160 1 CONTINUE
165 BIG=B
170 RETURN
180 END
```

BRAINY

```
100 SUBROUTINE BRAINY(NRA,NCA,LA,A,NRB,NCB,LB,B,C,LC)
110 DIMENSION A(NRA,NCA,LA),B(NCA,NCB,LB),C(NRA,NCB,LC)
120 COMPLEX A,B,C
130 * LC=LA+LB-1
140 CALL ZERO(NRA*NCB*LC,C)
150 DO 1 I=1,LA
160 DO 1 J=1,LB
170 K=I+J-1
180 DO 1 M=1,NRA
190 DO 1 N=1,NCB
200 DO 1 L=1,NCA
210 1 C(M,N,K)=C(M,N,K)+A(M,L,I)*B(L,N,J)
220 RETURN
230 END
```

CCAB

```
100 FUNCTION CCAB(X)
110 COMPLEX X
120 CCAB=(REAL(X)**2+AIMAG(X)**2)**0.5
130 RETURN
140 END
```

COHERE

```
100 SUBROUTINE COHERE(M1,NS,S,C)
110 DIMENSION S(M1,NS,NS),C(M1,NS,NS)
121 * EQUIVALENCE (S,C) IS ALLOWED
130 * SUBROUTINE COHERE COMPUTES THE MAGNITUDE AND PHASE ANGEL OF
140 * THE COHERENCY, AS WELL AS AUTOSPECTRA, EACH SCALED TO HAVE ITS LARGEST
150 * VALUE UNITY
160 DO 10 JP=2,NS
170 J=JP-1
180 DO 10 K=JP,NS
190 DO 10 I=1,M1
195 IF(S(I,J,J)*S(I,K,K).EQ.0.0)GO TO 10
200 CO=SQRT(ABS((S(I,J,K)**2+S(I,K,J)**2)/(S(I,J,J)*S(I,K,K))))
205 IF(ABS(S(I,J,K)).LT.1.0E-07)GO TO 101
210 PH=ATAN2(S(I,K,J),S(I,J,K))
220 102 C(I,J,K)=CO
230 10 C(I,K,J)=180.*PH/3.14159265
240 DO 20 J=1,NS
250 CALL MOVE(M1,S(1,J,J),C(1,J,J))
260 20 CALL NORMAG(M1,C(1,J,J))
270 RETURN
280 101 PH=SIGN(1.5707963,S(I,K,J))
290 GO TO 102
300 END
```

COQUAD

```
SUBROUTINE COQUAD(H,NS,M,N,W,R1,S,M1,LR)
* SUBROUTINE COQUAD COMPUTES THE MATRIX OF EMPIRICAL AUTOSPECTRA,
* COSPECTRA, AND QUADRATURE SPECTRA FROM  THE MULTI-CHANNEL
* AUTO CORRELATION FUNCTION
* NS=NUMBER OF TIME SERIES OR CHANNELS
* M=2**(N-1),M1=M+1=TIME LENGTH OF CORRLATION
* W(1)=1,W(M1)=0
* DIMENSION C(NM), WHERE NM IS NONDUMMY DIMENSION >=TWICE
* THE MAXIMUM LAG M
DIMENSION W(M),R1(LR,NS,NS),S(M1,NS,NS)
COMPLEX C(100)
DO 20 J=1,NS
DO 20 K=J,NS
DO 10 I=1,M
EVEN=R1(I,J,K)+R1(I,K,J)
ODD=R1(I,J,K)-R1(I,K,J)
R1(I,K,J)=W(I)*ODD
10 R1(I,J,K)=W(I)*EVEN
20 R1(1,J,K)=R1(1,J,K)*0.5
DO 40 J=1,NS
DO 40 K=J,NS
DO 1 I=1,M
1 C(I)=CMPLX(R1(I,J,K),R1(I,K,J))
DO 2 I=M1,M+M
2 C(I)=(0.,0.)
CALL NLOGN(N,C,-1.,M+M)
S(1,J,K)=H*REAL(C(1))
DO 3 I=2,M1
3 S(I,J,K)=H*(REAL(C(I))+REAL(C(M1+M1-I)))
IF(J-K) 7,40,40
7 DO 4 I=2,M1
4 S(I,K,J)=H*(REAL(C(I))-REAL(C(M1+M1-I)))
40 CONTINUE
* S(I,J,K) IS THE COSPECTRAL DENSITY OF THE JTH AND KTH
* CHANNEL EVALUATED AT LAG I-1 IF K>=J, AND EQUAL TO
* QUADSPECTRAL DENSITY EVALUATED AT LAG I-1 IF K<J, FOR J=1,M1
RETURN
END
```

CROSS

```
100 SUBROUTINE CROSS(LENX,X,Y,LR,R1,L)
110 * DIMENSION XX(NMAX),YY(NMAX)
120 * COMPLEX CX(NMAX),CY(NMAX),C(NMAX)
130 * NMAX IS A NONDUMMY DIMENSION>=2**(L+1), WHERE 2**L IS THE
140 * SMALLEST POWER OF 2 SUCH THAT LENX<=2**L
150 DIMENSION X(LENX),Y(LENX),R1(LR)
160 COMPLEX CX(2048),CY(2048),C(2048)
165 EQUIVALENCE (CX,C)
170 * LR<=LENX
180 LIBRARY "NLOGN"
185 L2=2**(L+1)
190 DO 1 J=1,LENX
195 CX(J)=CMPLX(X(J),0.0)
200 1 CY(J)=CMPLX(Y(J),0.0)
210 DO 2 J=LENX+1,L2
215 CX(J)=(0.0,0.0)
220 2 CY(J)=(0.0,0.0)
300 CALL NLOGN(L+1,CX,-1.0,L2)
310 CALL NLOGN(L+1,CY,-1.0,L2)
320 DO 4 J=1,L2
330 4 C(J)=CONJG(CX(J))*CY(J)
340 CALL NLOGN(L+1,C,1.0,L2)
350 DO 5 J=1,LR
360 5 R1(J)=REAL(C(J))/FLOAT(LENX)
370 * R1(J)=THE CROSS CORRELATION OF X AND Y EVALUATED AT
380 * LAG(J-1), J=1,2,...M+1
390 RETURN
400 END
```

DETREN

```
100 SUBROUTINE DETREN(NS,NV,SCANS)
110* NS=NUMBER OF SCANS
120* THE FOLLOWING IS USED TO DETREND A TIME SERIES
130* BY SUBTRACTING ITS LEAST SQUARES REGRESSION LINE
140* FROM EACH CHANNEL OF AN NV CHANNEL TIME SERIES
150 DIMENSION SCANS(NS,NV)
160 FNS=NS
170 TBAR=0.5*(FNS+1.0)
180 TSUMSQ=(FNS*(FNS+1.0)*(2.0*FNS+1.0))/6.0
190 DO 76 I2=1,NV
200 SUM=0.0
210 CRSPRO=0.0
220 DO 77 I1=1,NS
230 SUM=SUM+SCANS(I1,I2)
240 CRSPRO=CRSPRO+FLOAT(I1)*SCANS(I1,I2)
250 77 CONTINUE
260 FMEAN=SUM/FNS
270 BETA=(CRSPRO-FNS*TBAR*FMEAN)/(TSUMSQ-FNS*TBAR*TBAR)
280 DO 78 I1=1,NS
290 FREG=FMEAN+BETA*(FLOAT(I1)-TBAR)
300 SCANS(I1,I2)=SCANS(I1,I2)-FREG
310 78 CONTINUE
320 76 CONTINUE
330 RETURN
340 END
```

ESCAL

```
100 SUBROUTINE ESCAL(N,A,B)
110 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
120 DIMENSION A(N,N),B(N,N)
130 DO 5 I=1,N
140 DO 5 J=1,N
150 5 B(I,J)=0.0
160 B(1,1)=1./A(1,1)
170 IF(N.EQ.1)RETURN
180 DO 40 M=2,N
190 K=M-1
200 EK=A(M,M)
210 DO 10 I=1,K
220 DO 10 J=1,K
230 10 ED=EK-A(M,I)*B(I,J)*A(J,M)
240 B(M,M)=1./EK
250 DO 30 I=1,K
260 DO 20 J=1,K
270 20 B(I,M)=B(I,M)-B(I,J)*A(J,M)/EK
280 30 B(M,I)=B(I,M)
290 DO 40 I=1,K
300 DO 40 J=1,K
310 40 B(I,J)=B(I,J)+B(I,M)*B(M,J)*EK
320 RETURN
330 END
```

ESCALD

```
100 SUBROUTINE ESCALD(N,A,B,DET)
110 * SYMMETRIC MATRIX INVERSION BY ESCALATOR METHOD
120 * ALSO COMPUTES DETERMINANT OF A
130 DIMENSION A(N,N),B(N,N)
140 DO 5 I=1,N
150 DO 5 J=1,N
160 5 B(I,J)=0.0
170 B(1,1)=1./A(1,1)
180 DET=A(1,1)
190 IF(N.EQ.1)RETURN
200 DO 40 M=2,N
210 K=M-1
220 EK=A(M,M)
230 DO 10 I=1,K
240 DO 10 J=1,J
250 10 EK=EK-A(M,I)*B(I,J)*A(J,M)
260 DET=DET*EK
270 B(M,M)=1./EK
280 DO 30 I=1,K
290 DO 20 J=1,K
300 20 B(I,M)=B(I,M)-B(I,J)*A(J,M)/EK
310 30 B(M,I)=B(I,M)
320 DO 40 I=1,K
330 DO 40 J=1,K
340 40 B(I,J)=B(I,J)+B(I,M)*B(M,J)*EK
350 RETURN
360 END
```

FADDEJ

```
2400 SUBROUTINE FADDEJ(N,A,AINV,DET,ADJUG,P)
2410 LIBRARY "CCAB"
2420 DIMENSION A(N,N),AINV(N,N),ADJUG(N,N),P(N)
2430 COMPLEX A,AINV,DET,ADJUG,P
2440 COMPLEX PN
2450 NN=N*N
2460 CALL MOVEC(NN,A,AINV)
2470 DO 4 K=1,N
2480 P(K)=(0.0,0.0)
2490 DO 2 I=1,N
2500 2 P(K)=P(K)+AINV(I,I)
2510 P(K)=P(K)/FLOAT(K)
2520 IF(K.EQ.N)GO TO 5
2530 CALL MOVEC(N*N,AINV,ADJUG)
2540 DO 3 I=1,N
2550 3 ADJUG(I,I)=AINV(I,I)-P(K)
2560 4 CALL BRAINY(N,N,1,A,N,N,1,ADJUG,AINV,1)
2570 5 CALL MOVEC(N*N,ADJUG,AINV)
2580 E30=1.0E-30
2590 IF(CCAB(P(N)).LT.E30)GO TO 7
2600 DO 6 I=1,N
2610 DO 6 J=1,N
2620 PN=P(N)
2630 6 AINV(I,J)=AINV(I,J)/PN
2640 7 DET=P(N)
2650 IF(MOD(N,2).EQ.1)RETURN
2660 DET=-DET
2670 DO 8 I=1,N
2680 DO 8 J=1,N
2690 8 ADJUG(I,J)=-ADJUG(I,J)
2700 RETURN
2710 END
```

FAST

```
SUBROUTINE FAST(X1,Y1,M,N)
* SUBROUTINE FAST OBTAINS FINITE FOURIERTRANSFORMS OF THE PAIRSOF
*SERIES X AND Y
DIMENSION X(1024),Y(1024),Y1(N),X1(N)
DO 19 J=1,N
X(J)=X1(J)
19 Y(J)=Y1(J)
    N=2**M
    DO 1 L=1,M
    IMAX=2**(M-L)
    JDELT=2*IMAX
    ARG=6.2831853/FLOAT(JDELT)
    C=COS(ARG)
    S=SIN(ARG)
    U=1.0
    V=0.0
    DO 1 I=1,IMAX
    DO 2 J=I,N,JDELT
    K=J+IMAX
    XJ=X(J)+X(K)
    YJ=Y(J)+Y(K)
    XK=X(J)-X(K)
    YK=Y(J)-Y(K)
    X(K)=U*XK-V*YK
    Y(K)=U*YK+V*XK
    X(J)=XJ
2   Y(J)=YJ
    UT=C*U-S*V
    V=C*V+S*U
1   U=UT
    J=1
    NT=N/2
    IMAX=N-1
    DO 3 I=1,IMAX
    IF(I.GE.J) GO TO 5
    XT=X(J)
    X(J)=X(I)
    X(I)=XT
    YT=Y(J)
    Y(J)=Y(I)
    Y(I)=YT
5   K=NT
4   IF(K.GE.J) GO TO 3
    J=J-K
    K=K/2
    GO TO 4
3 J=J+K
DO 21 J=1,N
X1(J)=X(J)
21 Y1(J)=Y(J)
```

FAST (continued)

```
550  RETURN
560  END
```

FFT

```
      SUBROUTINE FFT(X,M)
      COMPLEX X(1024), U, W, T
        N=2**M
        NV2=N/2
        NM1=N-1
        J=1
        DO 7 I=1,NM1
        IF(I.GE.J) GO TO 5
        T=X(J)
        X(J)=X(I)
        X(I)=T
    5   K=NV2
    6   IF(K.GE.J ) GO TO 7
        J= J-K
        K=K/2
        GO TO 6
    7   J=J+K
        PI=3.14159265358979
        DO 20 L=1,M
        LE=2**L
        LE1=LE/2
        U=(1.0,0.0)
       W=CMPLX(COS(PI/FLOAT(LE1)), (+1.0)*SIN(PI/FLOAT(LE1)))
        DO 20 J=1,LE1
        DO 10 I=J,N,LE
        IP=I+LE1
        T=X(IP)*U
        X(IP) =X(I)-T
   10   X(I)=X(I)+T
   20   U=U*W
      RETURN
      END
```

FILTER

```
SUBROUTINE FILTER(LX,NWT,AW,WT,X,LY,Y,LN2N)
DIMENSION AW(NWT),WT(LX),X(LX),Y(LY)
* LY.LE.LX-NWT+1
* NWT=2*L+1,  2**LN2N.GE.LX
DO 1 J=1,NWT
1 WT(J)=AW(J)
DO 2 J=NWT+1,LX
2 WT(J)=0.0
CALL CROSS(LX,WT,X,LY,Y,LN2N)
FN=LX
DO 3 J=1,LY
3 Y(J)=FN*Y(J)
RETURN
END
```

FTFLTW

```
100 SUBROUTINE FTFLTW(A,NW1,NB2,AF,M,AT,NB1)
110 DIMENSION A(NW1),AT(NB1)
120 COMPLEX AF(NB2)
130 * NB2=2*NB=2**M.GE.NW1, NB1=NB+1
140 AF(1)=CMPLX(A(1),0.0)
150 DO 1 J=2,NW1
160 1 AF(J)=CMPLX(2.*A(J),0.0)
170 DO 2 J=NW1+1,NB2
180 2 AF(J)=(0.0,0.0)
190 CALL FFT(AF,M)
200 DO 3 J=1,NB1
210 3 AT(J)=REAL(AF(J))
220 RETURN
230 END
```

GENFLT

```
SUBROUTINE GENFLT(H,A,NW1,X,Y,L,Q,L2)
DIMENSION A(NW1),Q(L2),X(L),Y(L)
DATA P/6.2831853/
* NW1=NW+1, L2.GE.L1
L1=L+1
DO 5 K=1,L
5 X(K)=H*X(K)
DO 1 I=2,L
1 Q(I)=(Y(I)-Y(I-1))/(X(I)-X(I-1))
Q(1)=0.
Q(L1)=0.
DO 2 I=2,NW1
T=P*FLOAT(I-1)
TT=T*T
A(I)=0.
DO 3 J=2,L1
3 A(I)=A(I)+(Q(J-1)-Q(J))*COS(T*X(J-1))/TT
2 A(I)=(A(I)+(Y(L)*SIN(T*X(L))-Y(1)*SIN(T*X(1)))/T)*2
T=2.0*(Y(L)*X(L)-Y(1)*X(1))
DO 4 J=2,L
4 T=T-(Y(J)-Y(J-1))*(X(J)+X(J-1))
A(1)=T
RETURN
END
```

GENWTS

```
SUBROUTINE GENWTS(ID,L,WTS,NW2)
* NW2=2*L+1
DIMENSION WTS(NW2)
DATA PI/3.1415927/
IF(L.EQ.0) GO TO 21
FK=L/ID-2
FKK=FK+0.5
FL=L
DO 22 IS=1,L
S=IS
INDEX1=(L+1)+IS
HS=(1.0+COS(PI*S/FL))/(4.0*FL)
22 WTS(INDEX1)=HS*SIN(PI*FKK*S/FL)/SIN(PI*S/(2.0*FL))
DO 23 IS=1,L
INDEX1=L+1-IS
INDEX2=L+1+IS
23 WTS(INDEX1)=WTS(INDEX2)
WTS(L+1)=FKK/FL
RETURN
21 WTS(1)=1.0
RETURN
END
```

GFSORT

```
100 SUBROUTINE GFSORT(G,F,M)
110 DIMENSION G(M),F(M)
123 N=M
130 20 N=N/2
140 IF(N)30,40,30
150 30 K=M-N
160 J=1
170 41 I=J
180 49 L=I+N
190 IF(G(I)-G(L))50,60,60
200 50 B=G(I)
210 G(I)=G(L)
220 G(L)=B
230 B=F(I)
240 F(I)=F(L)
250 F(L)=B
260 I=I-N
270 IF(I-1)60,49,49
280 60 J=J+1
290 IF(J-K)41,41,20
300 40 RETURN
310 END
```

LEVNSN

```
100 SUBROUTINE LEVNSN(LR,R,A,S,M)
110 DIMENSION R(LR),A(LR)
120 V=R(1)
130 D=R(2)
140 A(1)=1.0
150 M=0
160 IF(LR.EQ.1)RETURN
170 DO 4 L=2,LR
180 A(L)=-D/V
190 S=V
200 IF(L.EQ.2)GO TO 2
210 L1=(L-2)/2
220 L2=L1+1
230 DO 1 J=2,L2
240 HOLD =A(J)
250 K=L-J+1
260 A(J)=A(J)+A(L)*A(K)
270 1 A(K)=A(K)+A(L)*HOLD
280 IF(2*L1.EQ.L-2)GO TO 2
290 A(L2+1)=A(L2+1)+A(L)*A(L2+1)
300 2 V=V+A(L)*D
305 M=M+1
307 PRINT,M,V
310 IF((S-V)/V -0.05)5,5,6
320 5 IF(M.GE.15)RETURN
330 6 IF(L.EQ.LR)RETURN
340 D=0.0
350 DO 4 I=1,L
360 K=L-I+2
370 4 D=D+A(I)*R(K)
380 RETURN
390 END
```

MAC

```
SUBROUTINE MAC(NS,LX,X,LR,R1,LXNS,L)
* SUBROUTINE MAC COMPUTES THE MULTICHANNEL AUTOCORRELATION OF
* THE NS CHANNEL TIME SERIES X
* LX=LENGTH OF EACH TIME SERIES IN EACH CHANNEL
* NS=NUMBER OF CHANNELS
* LR=DESIRED LENGTH OF CORRELATION, LR.LE.2**L
* L IS SMALLEST INTEGER SUCH THAT LX.LE.2**L
* LXNS=LX*NS
* NONDUMMY DIMENSION S(LF), WHERE LF.GE.ANTICIPATED LR
DIMENSION X(LXNS),R1(NS,NS,LR)
DIMENSION S(64)
DO 1  I=1,NS
I1=1+(I-1)*LX
DO 1  J=1,NS
J1=1+(J-1)*LX
CALL CROSS(LX,X(I1),X(J1),LR,S,L)
DO 1  K=1,LR
1 R1(J,I,K)=S(K)
RETURN
END
```

MACOR

```
SUBROUTINE MACOR(NS,LX,X,LR,R1,LXNS,LRNSNS,L)
* SUBROUTINE MACOR COMPUTES THE MULTI CHANNEL AUTO CORRELATION
* OF THE NS-CHANNEL TIME SERIES X
* LX= LENGTH OF EACH TIME SERIES IN EACH CHANNEL
* NS=NUMBER OF CHANNELS
* LR=DESIRED LENGTH OF CORRELATION, LR .LE. (LX-1)
* L IS THE SMALLEST INTEGER SUCH THAT LX.LE.2**L
* LXNX=LX*NS, LRNSNS=LR*NS*NS
DIMENSION X(LXNS),R1(LRNSNS)
DO 1 I=1,NS
I1=1+(I-1)*LX
DO 1 J=1,NS
J1=1+(J-1)*LX
IJ=1+LR*(I-1)+LR*NS*(J-1)
1 CALL CROSS(LX,X(I1),X(J1),LR,R1(IJ),L)
RETURN
END
```

MAINV

```
1500 SUBROUTINE MAINV(N,A,B)
1510 COMPLEX A,B,DET,ADJUG,P
1520 DIMENSION ADJUG(4,4),P(4)
1530 DIMENSION A(N,N),B(N,N)
1540 CALL FADDEJ(N,A,B,DET,ADJUG,P)
1550 RETURN
1560 END
```

MATMUL

```
100 SUBROUTINE MATMUL(N,A,B,C)
110 COMPLEX A(N,N),B(N,N),C(N,N)
120 DO 1 I=1,N
130 DO 1 J=1,N
140 C(I,J)=(0.0,0.0)
150 DO 1 K=1,N
160 1 C(I,J)=C(I,J)+A(I,K)*B(K,J)
170 RETURN
180 END
```

MOVE

```
SUBROUTINE MOVE(LX,X,Y)
DIMENSION X(LX),Y(LX)
DO 1 I=1,LX
1 Y(I)=X(I)
RETURN
END
```

MOVEC

```
 SUBROUTINE MOVEC(LX,X,Y)
DIMENSION X(LX),Y(LX)
COMPLEX X,Y
DO 1 I=1,LX
1 Y(I)=X(I)
RETURN
END
```

MULLEV

```
SUBROUTINE MULLEV(N,LF,R,A,AP,B,BP,VA,VB,DA,DB,CA,CB,M)
DIMENSION R(N,N,LF),A(N,N,LF),AP(N,N,LF),B( N,N,LF)
&,BP(N,N,LF),VA(N,N),VB(N,N),DA(N,N),DB(N,N),CA(N,N),CB(N,N)
CALL RZERO(N*N*LF,A)
CALL RZERO(N*N*LF,B)
DO 2 I=1,N
DO 1 J=1,N
VA(I,J)=R(I,J,1)
1 VB(I,J)=R(I,J,1)
A(I,I,1)=1.
2 B(I,I,1)=1.
CALL ESCALD(N,VA,CB,D)
M=0
DV=D
IF(LF.EQ.1)RETURN
DO 8 L=2,LF
CALL RZERO(N*N,DA)
DO 5 I=1,N
DO 4 LI=1,L
LD=L-LI+1
DO 4 K=1,N
DO 3 J=1,N
3 DA(I,J)=DA(I,J)-A(I,K,LI)*R(K,J,LD)
4 CONTINUE
DO 5 J=1,N
5 DB(J,I)=DA(I,J)
CALL SIMEQ(N,N,CA,VB,DA)
CALL SIMEQD(N,N,CB,VA,DB,DETVA)
IF((DV-DETVA)/DETVA-0.05)100,100,200
100 IF(M.GE.15)RETURN
200 DV=DETVA
M=M+1
CALL MOVE(N*N*L,A,AP)
CALL MOVE(N*N*L,B,BP)
DO 7 J=1,N
DO 7 K=1,N
DO 6 LI=1,L
LD=L-LI+1
DO 6 I=1,N
A(I,J,LI)=A(I,J,LI)+CA(I,K)*BP(K,J,LD)
6 B(I,J,LI)=B(I,J,LI)+CB(I,K)*AP(K,J,LD)
DO 7 I=1,N
VA(I,J)=VA(I,J)-CA(I,K)*DB(K,J)
7 VB(I,J)=VB(I,J)-CB(I,K)*DA(K,J)
8 CONTINUE
RETURN
END
```

NLOGN

```
      SUBROUTINE NLOGN(N,X,SIGN,LX)
*     NMAX=LARGEST VALUE OF N TO BE PROCESSED
*     NONDUMMY DIMENSION M(NMAX)
*     FOR EXAMPLE, IF NMAX=100, THEN
      DIMENSION M(100)
*     DIMENSION X(2**N)
      DIMENSION X(LX)
      COMPLEX X, WK, HOLD, Q
         DO 1 I=1,N
    1    M(I)=2**(N-I)
         DO 4 L=1,N
         NBLOCK=2**(L-1)
         LBLOCK=LX/NBLOCK
         LBHALF=LBLOCK/2
         K=0
         DO 4 IBLOCK=1,NBLOCK
         FK=K
         FLX=LX
         V=SIGN*6.2831853*FK/FLX
         WK=CMPLX(COS(V),SIN(V))
         ISTART=LBLOCK*(IBLOCK-1)
         DO 2 I=1,LBHALF
         J=ISTART+I
         JH=J+LBHALF
         Q=X(JH)*WK
         X(JH)=X(J)-Q
         X(J)=X(J)+Q
    2    CONTINUE
         DO 3 I=2,N
         II=I
         IF(K.LT.M(I)) GO TO 4
    3    K=K-M(I)
    4    K=K+M(II)
         K=0
         DO 7 J=1,LX
         IF(K.LT.J) GO TO 5
         HOLD=X(J)
         X(J)=X(K+1)
         X(K+1)=HOLD
    5    DO 6 I=1,N
         II=I
         IF(K.LT.M(I)) GO TO 7
    6    K=K-M(I)
    7    K=K+M(II)
         IF(SIGN.LT.0.0) RETURN
         DO 8 I=1,LX
    8    X(I)=X(I)/FLX
      RETURN
      END
```

NORMAG

```
100 SUBROUTINE NORMAG(LX,X)
110 DIMENSION X(LX)
125 B=0.0
130 DO 10 I=1,LX
140 10 B=AMAX1(ABS(X(I)),B)
145 IF(B.EQ.0.0)RETURN
150 DO 20 I=1,LX
160 20 X(I)=X(I)/B
170 RETURN
180 END
```

```
OLDPLO

100 SUBROUTINE OLDPLO(G2,F1,M)
110 DIMENSION G2(M),F1(M)
111 DIMENSION F(64),G(64)
119 CHARACTER BLANK(63)/63*" "/
121 CHARACTER STAR/"*"/
130 ROUND(X)=X+.5
132 DO 15 I=1,M
133 G(I)=G2(I)
134 15 F(I)=F1(I)
140 16 FG=(BIG(G,M)-SMALL(G,M))/42.
145 IF(FG.LE.1.0E-07)GO TO 998
150 DO 1 K=1,M
160 G(K)=G(K)/FG
170 1 F(K)=2.0*F(K)
180 CALL GFSORT(G,F,M)
190 DO 19 K=1,5
200 19 WRITE(0,23)
210 23 FORMAT(1H )
220 G1=G(1)*FG
230 WRITE(0,100)G1
240 GLAST=G(1)
250 DO 2 J=1,M
260 LGDIF=ROUND(GLAST-G(J))
270 IF(LGDIF)8,8,6
280 6 DO 7 L=1,LGDIF
290 GG=(GLAST-FLOAT(L))*FG
300 7 WRITE(0,100)GG
310 100 FORMAT(1H , E9.2," ")
320 8 LFDIF=ROUND(F(J)+1.)
330 WRITE(0,101)(BLANK(K),K=1,LFDIF),STAR
340 101 FORMAT(1H+,10X,64A1)
350 2 GLAST=G(J)
360 WRITE(0,200)
370 200 FORMAT(10X,32(' I'))
380 WRITE(0,210)
390 210 FORMAT(10X,' 0 1 2 3 4 5 6 7 8 9  11  13  15  17  19  21',
400 &'  23  25  27  29  31')
410 RETURN
420 998 WRITE(0,104)G(1)
430 104 FORMAT('ALL VALUES ARE EQUAL TO',E9.2)
440 RETURN
500 END
```

PDP

```
100 *10
110 FL2IND,0
120 FL3IND,0
130 *200
140 STA
150 CLZE
160 CLA
170 TAD MTICKS
180 CLAB
190 CLA
200 TAD K5500
210 CLOE
220 CLA
230 STA
240 DCA FL2IND
250 STA
260 DCA FL3IND
270 DCA TALLY
280 LOOP, CLSK
290 JMP .-1
295 CLSA
300 JMS SAMP
310 ISZ TALLY
320 JMP LOOP
330 HLT
340 SAMP, 0
350 CLA
360 TAD K2000
370 JMS ADCONV
380 6221 / CDF2
390 DCA I FL2IND
400 TAD K2001
410 JMS ADCONV
420 6231 /CDF3
430 DCA I FL3IND
440 JMP I SAMP
450 ADCONV, 0
460 ADSC
470 ADCV
480 ADSF
490 JMP .-1
500 ADRB
510 JMP I ADCONV
520 K2001, 2001
530 K2000, 2000
540 TALLY, 0
550 MTICKS, -1415
560 K5500, 5500
570 *300
580 DISP, LAS
```

PDP (continued)

```
590 SMA
600 JMP .+3
610 6231 /CDF3
620 JMP .+2
630 6221 /CDF2
640 CLA
650 DCA TALLY
652 TAD K2000
654 6552
656 CLA
658 6552
660 TAD I TALLY
670 6551
680 CLA
690 ISZ TALLY
700 JMP .-4
710 JMP DISP
720 $
```

PLOT

```
100 SUBROUTINE PLOT(G2,F1,M)
110 DIMENSION G2(M),F1(M)
111 DIMENSION F(64),G(64)
119 CHARACTER BLANK(63)/63*" "/
121 CHARACTER STAR/"*"/
130 ROUND(X)=X+.5
132 DO 15 I=1,M
133 G(I)=G2(I)
134 15 F(I)=F1(I)
140 16 FG=(BIG(G,M)-SMALL(G,M))/42.
145 IF(FG.LE.1.0E-07)GO TO 998
150 DO 1 K=1,M
160 G(K)=G(K)/FG
170 1 F(K)=2.0*F(K)
180 CALL GFSORT(G,F,M)
190 DO 19 K=1,5
200 19 WRITE(0,23)
210 23 FORMAT(1H )
220 G1=G(1)*FG
230 WRITE(0,100)G1
240 GLAST=G(1)
250 DO 2 J=1,M
260 LGDIF=ROUND(GLAST-G(J))
270 IF(LGDIF)8,8,6
280 6 DO 7 L=1,LGDIF
290 GG=(GLAST-FLOAT(L))*FG
300 7 WRITE(0,100)GG
310 100 FORMAT(1H , E9.2," ")
320 8 LFDIF=ROUND(F(J)+1.)
330 WRITE(0,101)(BLANK(K),K=1,LFDIF),STAR
340 101 FORMAT(1H+,10X,64A1)
350 2 GLAST=G(J)
360 WRITE(0,200)
370 200 FORMAT(10X,32(' I'))
380 WRITE(0,210)
390 210 FORMAT(10X,' 0 1 2 3 4 5 6 7 8 9  11  13  15  17  19  21',
400 &'  23  25  27  29  31')
410 RETURN
420 998 WRITE(0,104)G(1)
430 104 FORMAT('ALL VALUES ARE EQUAL TO',E9.2)
440 RETURN
500 END
```

PLOTTR

```
100 SUBROUTINE PLOTTR(Y1,X,M)
110 DIMENSION Y1(M),X(M),Y(128)
126 CHARACTER LINE(63)/63*" "/,BLAN/" "/,STAR/"*"/,DOT/"."/
129 DO 99 I=1,M
130 99 Y(I)=Y1(I)
131 SMALL=Y(1)
140 BIG=SMALL
150 DO 40 I=2,M
160 VALUE=Y(I)
170 IF(VALUE-BIG)20,20,10
180 10 BIG=VALUE
190 GO TO 40
200 20 IF(VALUE-SMALL)30,40,40
210 30 SMALL=VALUE
220 40 CONTINUE
230 IF(ABS(BIG)-ABS(SMALL))50,60,60
240 50 SCALE=ABS(SMALL)/31.
250 GO TO 70
260 60 SCALE=ABS(BIG)/31.
270 70 IF(BIG-SMALL)100,80,100
280 80 WRITE(0,90)BIG
290 90 FORMAT(' NO GRAPH CAN BE DRAWN SINCE ALL VALUES ARE ',E15.7)
300 RETURN
310 100 WRITE(0,110)BIG,SMALL,SCALE
320 110 FORMAT (' LARGEST VALUE IS ',E15.7,/,'SMALLEST VALUE IS ',
330 &E15.7,/,' MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR ',E15.7)
340 DO 19 K=1,5
350 19 WRITE(0,23)
360 23 FORMAT(1H )
370 DO 130 I=1,M
380 130 Y(I)=Y(I)/SCALE
390 WRITE(0,210)
400 210 FORMAT(9X,'-30-27-24-21-18-15-12 -9 -6 -3  0  3  6  9 12',
410 &' 15 18 21 24 27 30')
420 WRITE(0,120)
430 120 FORMAT(11X,21('I  '))
440 DO 131 I=1,63
450 131 LINE(I)=BLAN
460 DO 1000 I=1,M
470 VALUE=Y(I)
480 INDEX=32.+VALUE
481 INDEX=MAX0(INDEX,1)
482 INDEX=MIN0(INDEX,63)
490 LINE(32)=DOT
500 LINE(INDEX)=STAR
510 WRITE(0,180)X(I),LINE
520 LINE(INDEX)=BLAN
530 180 FORMAT(F7.2,3X,63A1)
540 1000 CONTINUE
550 RETURN
```

PLOTTR (continued)

```
560 END
```

REMAV

```
SUBROUTINE REMAV(LY,Y)
DIMENSION Y(LY)
S=0.0
DO 10 I=1,LY
10 S=S+Y(I)
AV=S/FLOAT(LY)
DO 20 I=1,LY
20 Y(I)=Y(I)-AV
RETURN
END
```

RZERO

```
SUBROUTINE RZERO(LX,X)
DIMENSION X(LX)
DO 1 I=1,LX
1 X(I)=0.0
RETURN
END
```

SIMEQ

```
SUBROUTINE SIMEQ(M,N,A,B,C)
* NMAX=LARGEST VALUE OF N TO BE PROCESSED
* NONDUMMY DIMENSION S(NMAX,NMAX)
* FOR EXAMPLE, IF NMAX=4 THEN
DIMENSION S(4,4)
DIMENSION A(M,N),B(N,N),C(M,N)
CALL MOVE(N*N,B,S)
CALL ESCAL(N,S,B)
DO 1 I=1,M
DO 1 J=1,N
A(I,J)=0.0
DO 1 K=1,N
1 A(I,J)=A(I,J)+C(I,K)*B(K,J)
CALL MOVE(N*N,S,B)
RETURN
END
```

SIMEQD

```
SUBROUTINE SIMEQD(M,N,A,B,C,D)
* NMAX=LARGEST VALUE OF N TO BE PROCESSED
* NONDUMMY DIMENSION A(NMAX,NMAX)
* FOR EXAMPLE, IF NMAX=4, THEN
DIMENSION S(4,4)
DIMENSION A(M,N),B(N,N),C(M,N)
CALL MOVE(N*N,B,S)
CALL ESCALD(N,S,B,D)
DO 1 I=1,M
DO 1 J=1,N
A(I,J)=0.0
DO 1 K=1,N
1 A(I,J)=A(I,J)+C(I,J)*B(K,J)
CALL MOVE(N*N,S,B)
RETURN
END
```

SMALL

```
100 FUNCTION SMALL(A,M)
110 DIMENSION A(M)
120 S=A(1)
130 DO 1 K=2,M
140 IF(A(K)-S)2,1,1
150 2 S=A(K)
160 1 CONTINUE
165 SMALL=S
170 RETURN
180 END
```

TRANS

```
SUBROUTINE TRANS(P,IDIMP,Y,N,NV,NB,LOG2NS)
DIMENSION Y(1024,NV)
DIMENSION P(IDIMP)
M=LOG2NS
 DO 1 I=1,NV-1,2
1 CALL FAST(Y(1,I),Y(1,I+1),M,N)
NT=N/2
DO 8 I=1,NV,2
DO 8 J=2,NT
K=N-J+2
T1=(Y(J,I+1)-Y(K,I+1))
T2=(Y(K,I)-Y(J,I))
Y(J,I)=Y(J,I)+Y(K,I)
Y(J,I+1)=Y(J,I+1)+Y(K,I+1)
Y(K,I)=T1
8 Y(K,I+1)=T2
DO 17 I=1,NV
Y(1,I)=2.0*Y(1,I)
17 Y(NT+1,I)=(2.0)*Y(NT+1,I)
NE=N/(4*NB)
IIMIN=NE+1
IIMAX=NT-NE
IISTEP=NE+NE
L=1
DO 2 I=1,NV
DO 2 J=I,NV
P(L)=Y(1,I)*Y(1,J)
P(L+1)=0.0
DO 3 K=2,IIMIN
M=N-K+2
3 P(L)=P(L)+2.0*(Y(K,I)*Y(K,J)+Y(M,I)*Y(M,J))
L=L+2
DO 4 II=IIMIN,IIMAX,IISTEP
P(L)=0.0
P(L+1)=0.0
KMAX=II+IISTEP
DO 5 K=II,KMAX
M=N-K+2
P(L)=P(L)+Y(K,I)*Y(K,J)+Y(M,I)*Y(M,J)
5 P(L+1)=P(L+1)+Y(K,I)*Y(M,J)-Y(M,I)*Y(K,J)
4 L=L+2
P(L)=Y(NT+1,I)*Y(NT+1,J)
P(L+1)=0.0
KMIN=NT+1-NE
DO 6 K=KMIN,NT
M=N-K+2
6 P(L)=P(L)+2.0*(Y(K,I)*Y(K,J)+Y(M,I)*Y(M,J))
2 L=L+2
RETURN
END
```

UNPACK

```
100 LET I=0
110 DIM A(4096),S(13)
120 FILE #1: "PUB"
130 FOR J = 1 TO LOF(#1)
140 READ #1 : S$
150 CHANGE S$ TO S
160 FOR K= 0 TO 3
170 LET I=I+1
180 LET A(I)=16*S(3*K+1)+INT(S(3*K+2)/16)
190 LET I=I+1
200 LET A(I)=256*MOD(S(3*K+2),16)+S(3*K+3)
210 NEXT K
220 NEXT J
230 SCRATCH #1
240 FOR J= 1 TO I
250 WRITE #1: A(J)
260 NEXT J
270 END
```

WPARZ

```
100 SUBROUTINE WPARZ(M,W)
110 DIMENSION W(M)
129 ** M IS AN EVEN INTEGER
130 DO 1 J=1,M/2 +1
140 1 W(J)=1.-6.*(FLOAT(J-1)/FLOAT(M))**2+6.*(FLOAT(J-1)/FLOAT(M))**3
150 DO 2 J=M/2+2,M
160 2 W(J)=2.*(1.-FLOAT(J-1)/FLOAT(M))**3
170 RETURN
180 END
```

ZERO

```
SUBROUTINE ZERO(LX,X)
DIMENSION X(LX)
COMPLEX X
IF(LX.LE.0)RETURN
DO 1 I=1,LX
1 X(I)=(0.0,0.0)
RETURN
END
```

LPSFLT1

```
100 DIMENSION WT(2095),WTS(49),AI(2095),BI(2047)
110 LIBRARY "NLOGN","CROSS","GENWTS","FILTER"
120 OPENFILE 1,"PUB","NUMERIC"
130 READ(1) (AI(J),J=1,2095)
140 CALL GENWTS(2,24,WTS,49)
150 CALL FILTER(2095,49,WTS,WT,AI,2047,BI,12)
170 OPENFILE 2,"NPUB","NUMERIC"
180 WRITE(2)BI
250 STOP
260 END
```

LPSFLT2

```
100 DIMENSION WT(2095),WTS(49),A2(2095),B2(2047)
110 LIBRARY "NLOGN","CROSS","GENWTS","FILTER"
140 CALL GENWTS(2,24,WTS,49)
190 OPENFILE 3,"CH2","NUMERIC"
200 READ(3) (A2(J),J=1,2095)
210 CALL FILTER(2095,49,WTS,WT,A2,2047,B2,12)
230 OPENFILE 4,"NCH2","NUMERIC"
240 WRITE(4)B2
250 STOP
260 END
```

SPECTRUM

```
100 DIMENSION Y(320),R1(300),W(32),G(33),F(33),GH(33),TLAG(300)
110 LIBRARY "RSPECT","REMAV","NLOGN","CROSS","WPARZ","SPECT"
120 &,"SMOOTH","PLOT","GFSORT","BIG","SMALL","PLOTTR"
130 M=32
140 LY=320
150 OPENFILE 3,"EEGDAT","NUMERIC"
160 READ(3)Y
170 CALL REMAV(LY,Y)
180 CALL CROSS(320,Y,Y,70,R1,9)
190 CALL WPARZ(M,W)
200 H=1./64.
210 M1=M+1
220 N=6
230 CALL SPECT(H,M,N,W,R1,G,M1)
240 DO 1 J=1,33
250 1 F(J)=J-1
260 CALL PLOT(G,F,M)
270 WRITE(0,100)(F(J),G(J),J=1,33)
275 100 FORMAT(1H0,F5.2,4X,E9.2/)
280 DO 2 J=1,70
290 2 TLAG(J)=J-1
300 CALL PLOTTR(R1,TLAG,50)
310 WRITE(0,200)(TLAG(J),R1(J),J=1,70)
320 200 FORMAT(1H0,F7.1,4X,E9.2/)
330 PRINT,Y
340 STOP
350 END
```

```
SUBROUTINE SPECT(H,M,N,W,R1,G,M1)
DIMENSION W(M),R1(M1),G(M1)
COMPLEX C(100)
* M1=M+1,W(1)=1,2*M=2**N
* DIMENSION C(NM) WHERE NM IS NONDUMMY DIMENSION MORE THAN
* TWICE MAXIMUN LAG M
C(1)=CMPLX(R1(1),0.)
DO 1 J=2,M
1 C(J)=CMPLX(2.*W(J)*R1(J),0.)
DO 2 J=M+1,M+M
2 C(J)=(0.,0.)
CALL NLOGN(N,C,-1.,M+M)
DO 3 J=1,M1
3 C(J)=2.*H*REAL(C(J))
RETURN
END
```

AD-A032 719 NAVAL ACADEMY ANNAPOLIS MD F/G 9/3

MATHEMATICAL-STATISTICAL AND DIGITAL COMPUTER ANALYSIS OF TIME --ETC(U)

MAY 76 S J RAHER

UNCLASSIFIED USNA-TSPR-81 NL

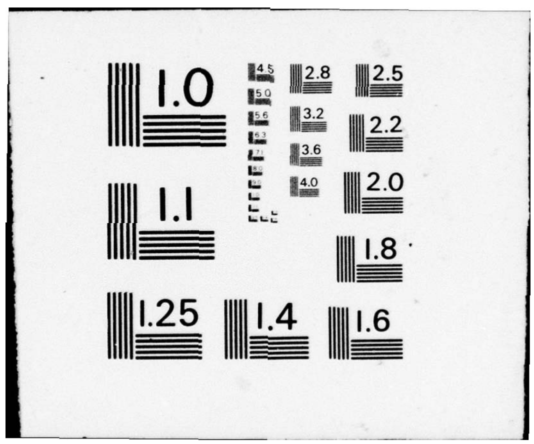
2 OF 2

ADA032719

END

DATE FILMED

2 -77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

```
DIMENSION SPECT(LS)
MM=LS-1
A=.54*SPECT(1)+.46*SPECT(2)
B=.54*SPECT(LS)+.46*SPECT(MM)
SJ=SPECT(1)
SK=SPECT(2)
DO 10 J=2,MM
SI=SJ
SJ=SK
SK=SPECT(J+1)
10 SPECT(J)=.54*SJ+.23*(SI+SK)
SPECT(1)=A
SPECT(LS)=B
RETURN
END
```

```
100 SUBROUTINE RSPECT(H,M,N,R1,G,M1)
110 DIMENSION R1(M1),G(M1)
120 COMPLEX C(100)
130 * M1=M+1; 2*M=2**N
140 * DIMENSION C(NM), WHERE NM IS NONDUMMY DIMENSION >= TWICE
150 * THE MAXIMUM LAG M.
160 C(1)=CMPLX(R1(1),0.)
170 DO 1 J=2,M
180 1 C(J)=CMPLX(2.*R1(J),0.)
190 C(M1)=CMPLX(R1(M1),0.)
200 DO 2 J=M+2,M+M
210 2 C(J)=(0.,0.)
220 CALL NLOGN(N,C,-1.,M+M)
230 DO 3 J=1,M1
240 3 G(J)=2.*H*REAL(C(J))
250 RETURN
260 END
```

TRUSPEC

```
100 *NV=NUMBER OF CHANNELS
110 * NB=NUMBER OF FREQUENCY BANDS(A POWER OF2)
120 LIBRARY "OLDPL ","GFSORT","BIG","SMALL","CCAB","FFT","SPKFLT","GENFLT","FTFLTW"
130 REAL F(33),W2(33,4,4,4),SP(33),AA(51),AT(33),Q(9),XF(8),YF(8),A(4,5),D(4,5),W(4,5),C(4)
140 COMPLEX E,S113,S223,S123,S112,S332,S132,S221,S331,S231,S(33,4,4),AF(64),SYY(33),CS
150 CHARACTER CH(4)/"1","2","3","4"/
160 DATA NV,NB,H/4,32,0.015625/
170 DATA NW1,L,L1,NB2,M/51,8,9,64,6/
180 DATA C(1),C(2),C(3),C(4)/0.4,0.1,0.1,0.1/
190 DATA XF/4.,5.,7.,8.,12.,13.,15.,16./
200 DATA YF/0.,1.,1.,0.,0.,0.5,0.5,0./
210 DATA A(1,1),D(1,1),W(1,1)/2.,0.,1./
220 DATA A(1,2),D(1,2),W(1,2)/4.,6.,1./
230 DATA A(1,3),D(1,3),W(1,3)/3.,10.,1./
240 DATA A(1,4),D(1,4),W(1,4)/4.5,14.,2./
250 DATA A(1,5),D(1,5),W(1,5)/4.,29.,1./
260 DATA A(2,1),D(2,1),W(2,1)/0.1,10.,1./
270 DATA A(2,2),D(2,2),W(2,2)/0.7,20.,1./
280 DATA A(3,1),D(3,1),W(3,1)/0.8,3.,1./
290 DATA A(3,2),D(3,2),W(3,2)/0.2,25.,1./
300 DATA A(4,1),D(4,1),W(4,1)/.25,12.,1./
310 DATA A(4,2),D(4,2),W(4,2)/0.,1.,1./
320 NB1=NB+1
330 N=NV
340 DO 1 J=1,NV
350 DO 1 K=1,NV
360 DO 1 I=1,NB1
370 1 S(I,J,K)=(0.,0.)
380 DO 2 K=1,5
390 CALL SPKFLT(H,A(1,K),D(1,K),W(1,K),NW1,AA)
400 CALL FTFLTW(AA,NW1,NB2,AF,M,AT,NB1)
410 DO 2 I=1,NB1
420 2 S(I,1,1)=S(I,1,1)+CMPLX(ABS(AT(I))**2,0.)
430 DO 10 I=1,NB1
440 10 S(I,1,1)=H*S(I,1,1)+H*C(1)**2
450 CALL GENFLT(H,AA,NW1,XF,YF,L,Q,L1)
460 CALL FTFLTW(AA,NW1,NB2,AF,M,AT,NB1)
470 DO 3 I=1,NB1
480 DO 3 J=2,4
490 S(I,1,J)=S(I,1,1)*AT(I)
500 3 SYY(I)=S(I,1,1)*ABS(AT(I))**2
510 DO 4 I=1,NB1
520 CS=SYY(I)
530 S(I,2,4)=CS
540 S(I,3,4)=CS
550 S(I,3,2)=CS
560 S(I,4,2)=CS
570 S(I,4,3)=CS
580 4 S(I,2,3)=CS
590 DO 5 N=2,4
600 DO 5 K=1,2
610 CALL SPKFLT(H,A(N,K),D(N,K),W(N,K),NW1,AA)
620 CALL FTFLTW(AA,NW1,NB2,AF,M,AT,NB1)
630 DO 5 I=1,NB1
640 5 S(I,N,N)=S(I,N,N)+CMPLX(ABS(AT(I))**2,0.)
650 DO 6 N=2,4
660 DO 6 I=1,NB1
670 S(I,N,N)=SYY(I)+H*(S(I,N,N)+C(N)**2)
```

TRUSPEC WAS USED TO COMPUTE THE THEORETICAL SPECTRA OF SIMULATED TIME-SERIES DATA.

SINCE THE DATA WAS OBTAINED BY PASSING WHITE NOISE THROUGH DIGITAL FILTERS, AND THE WHITE NOISE PROCESS WAS OBTAINED BY USING A FAULTY RANDOM NUMBER GENERATOR, THE ESTIMATED COHERENCE SPECTRA AND ESTIMATED PARTIAL COHERENCE SPECTRA DIFFERED SOMEWHAT FROM THE CORRESPONDING THEORETICAL QUANTITIES.

```
6 CONTINUE
DO 7 I=1,NB1
DO 7 J=2,4
7 S(I,J,1)=S(I,1,J)
DO 201 J=1,NV-1
DO 201 K=J+1,NV
DO 201 I=1,NB1
CSS=CCAB(S(I,J,J))*CCAB(S(I,K,K))
  IF(CSS-1.0E-07) 17,18,18
17 S(I,K,J)=(0.0,0.0)
GO TO 201
18 S(I,K,J)=CMPLX(CCAB(S(I,J,K))**2/CSS,0.0)
201 CONTINUE
DO 202 J1=1,NV-2
DO 202 J2=J1+1,NV-1
DO 202 J3=J2+1,NV
DO 202 I=1,NB1
 RES=REAL(S(I,J3,J3))
  S113=S(I,J1,J1)*(1.0-S(I,J3,J1))
S223=S(I,J2,J2)*(1.-S(I,J3,J2))
IF(RES-1.0E-07) 901,902,902
901 S123=S(I,J1,J2)
GO TO 903
902 S123=S(I,J1,J2)-S(I,J1,J3)*CONJG(S(I,J2,J3))/RES
903 IF(CCAB(S113)*CCAB(S223)-1.0E-07) 601,602,602
601 W2(I,J1,J2,J3)=0.0
GO TO 202
 602 W2(I,J1,J2,J3)=(CCAB(S123)**2)/(CCAB(S113)*CCAB(S223))
RES=REAL(S(I,J2,J2))
S112=S(I,J1,J1)*(1.0-S(I,J2,J1))
S332=S(I,J3,J3)*(1.-S(I,J3,J2))
IF(RES-1.0E-07) 1001,1002,1002
1001 S132=S(I,J1,J3)
GO TO1003
1002 S132=S(I,J1,J3)-S(I,J1,J2)*S(I,J2,J3)/RES
1003 IF(CCAB(S112)*CCAB(S332)-1.0E-07) 701,702,702
701 W2(I,J1,J3,J2)=0.0
GO TO 202
702 W2(I,J1,J3,J2)=(CCAB(S132)**2)/(CCAB(S112)*CCAB(S332))
RES=REAL(S(I,J1,J1))
S221=S(I,J2,J2)*(1.-S(I,J2,J1))
S331=S(I,J3,J3)*(1.-S(I,J3,J1))
IF(RES-1.0E-07) 1101,1102,1102
1101 S231=S(I,J2,J3)
GO TO 1103
1102 S231=S(I,J2,J3)-S(I,J1,J3)*CONJG(S(I,J1,J2))/RES
1103 IF(CCAB(S221)*CCAB(S331)-1.0E-07) 801,802,802
801 W2(I,J2,J3,J1)=0.0
GO TO 202
    802 W2(I,J2,J3,J1)=(CCAB(S231)**2)/(CCAB(S221)*CCAB(S331))
202 CONTINUE
CONTINUE
DO 71 J=1,NB1
71 F(J)=J-1
DO 500 J=1,NV-1
DO 500 K=J+1,NV
WRITE(0,300)CH(J),CH(K)
300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
```

```
DO 319 I=1,NB1
  SP(I)=REAL(S(I,K,J))
319 CONTINUE
 CALL OLDPLO(SP,F,NB)
  WRITE(0,102)
102 FORMAT(5(1H ,/))
DO 500 L=1,NV
IF((J-L)*(K-L))418,500,418
 418  DO 512 I=1,NB1
 SP(I)=W2(I,J,K,L)
512 CONTINUE
WRITE(0,301) CH(J),CH(K),CH(L)
301 FORMAT(' PARTIAL COHERENCE BETWEEN CHANNELS ',A1,' AND ',
&A1,/,' AFTER THE INFLUENCE OF CHANNEL ',A1,' HAS BEEN REMOVED')
CALL OLDPLO(SP,F,NB)
WRITE(0,102)
500 CONTINUE
DO 417 J=1,NV
WRITE(0,317)CH(J)
317 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
IX=2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2)-1
DO 416 I=1,NB1
416   SP(I)=REAL(S(I,J,J))
  CALL OLDPLO(SP(1),F,NB)
417 WRITE(0,102)
STOP
END
```

WNSSIM

```
100 DIMENSION X(1000,2),Z(1000)
110 LIBRARY "WNOISE","FORTLIB***:FSAVFL"
120 DATA LZ/1000/
130   DO 1 J=1,2
140   CALL WNOISE(LZ,Z,8.0)
150   DO1 I=1,1000
160 1  X(I,J)=Z(I)
165  CALL FSAVFL("WNSDAT"," ",I)
170 OPENFILE 2,"WNSDAT","NUMERIC"
180  WRITE(2) ((X(I,J),I=1,1000),J=1,2)
190 STOP
200 END
```

```
110 * NV=NUMBER OF CHANNELS
120 * NB=NUMBER OF FREQUENCY BANDS(A POWER OF 2)
140 * JSCANS  IS ALSO A POWER OF 2
150 * SR=SAMPLING RATE=1/H
160 * X=INPUT SERIES(ARRAY)
170 * P=ARRAY FOR STORING CROSS SPECTRA
180 DIMENSION X(1024,2),P(198),S(132),C(132),F(33)
185 DIMENSION S1(33,2,2)
190 CHARACTER CH(2)/"1","2"/
200 EQUIVALENCE (S,C)
201 EQUIVALENCE (S,S1)
210 DATA NS,NV,NB,JSCANS,SR,PI/1000,2,32,1024,64.,3.141592657/
215 LIBRARY "FAST","TRANS","MOVE","NORMAG","COHERE","PLOT","GFSORT"
216 &,"BIG","SMALL"
220 OPENFILE 2,"WNSDAT","NUMERIC"
230 READ(2)((X(J,I),J=1,NS),I=1,NV)
240 * DETREND THE SERIES BY SUBTRACTING FROM EACH SERIES ITS
250 * LEAST SQUARES LINEAR REGRESSION LINE
260 FNS=NS
270 TBAR=0.5*(FNS+1.0)
280 TSUMSQ=(FNS*(FNS+1.0)*(FNS+FNS+1.0))/6.0
290 DO 76 I2=1,NV
300 SUM=0.0
310 CRSPRO=0.0
320 DO 77 I1=1,NS
330 SUM=SUM+X(I1,I2)
340 77 CRSPRO =CRSPRO+FLOAT(I1)*X(I1,I2)
350 FMEAN=SUM/FNS
360 BETA=(CRSPRO-FNS*TBAR*FMEAN)/(TSUMSQ-FNS*TBAR*TBAR)
370 DO 78                    I1=1,NS
380 FREG=FMEAN+BETA*(FLOAT(I1)-TBAR)
390 78              X(I1,I2)=X(I1,I2)-FREG
400 76 CONTINUE
410 * WINDOW EACH SERIES WITH A COSINE TAPER
420 IR=NS/10
430 R=IR
440 DO 79 I1=1,IR
450 FI1=I1
455 FINT=FI1-0.5
460 WINDOW=0.5*(1.0-COS(PI*FINT/R))
470 I3=NS+1-I1
480 DO 80 I2=1,NV
490 X(I1,I2)=WINDOW*X(I1,I2)
500 80 X(I3,I2)=WINDOW*X(I3,I2)
510 79 CONTINUE
520 LOG2NS=0
530 NSCANS=1
540 54 IF(NS.LE.NSCANS)GO TO 55
550 LOG2NS=LOG2NS+1
560 NSCANS=NSCANS+NSCANS
570 GO TO 54
580 55 IF(NS.EQ.NSCANS)GO TO 74
590 * ZERO OUT THE LAST PART OF EACH SERIES IF THE NUMBER OF
600 * SCANS IS NOT A POWER OF 2
610 I1BEGN=NS+1
620 DO 75 I1=I1BEGN,NSCANS
630 DO 75 I2=1,NV
```

```
75 X(I1,I2)=0.0
74 CONTINUE
IF(MOD(NV,2))70,82,70
* IF THE NUMBER OF SERIES IS ODD, FILL A DUMMU SERIES WITH ZEROS
70 NV1=NV+1
DO 83 I1=1,NSCANS
83 X(I1,NV1)=0.0
GO TO 85
82 NV1=NV
85 CONTINUE
IDIMP=NV1*(NV1+1)*(NB+1)
 CALL TRANS(P,IDIMP,X,NSCANS,NV1,NB,LOG2NS)
* CROSS SPECTRUM ESTIMATES ARE IN ARRAY P
* THE CROSS SPECTRUM ESTIMATES IN ARRAY P ARE SCALED BY
* MULTIPLYING BY C
WNDPWR=FNS-1.25*R
FSCANS=NSCANS
FNB=NB
FD=FSCANS/(FNB+FNB)
C1=0.25/(SR*(FD+1.0)*WNDPWR)
IROWSP=NB+NB+2
ICOLSP=(NV1*(NV1+1))/2
ISIZEP=IROWSP*ICOLSP
DO 95 I1=1,ISIZEP
95 P(I1)=C1*P(I1)
NB1=NB+1
DO 1000 J=1,NV1
DO 1000 K=J,NV1
DO 1000 I=1,NB1
S1(I,J,K)=P(2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2+K-J)+I+I-1)
IF(J-K)99,1000,1000
99 S1(I,K,J)=P(2*NB1*(NV1*(J-1)-((J-1)*(J-2))/2+K-J)+I+I)
1000 CONTINUE
CALL COHERE(NB1,NV1,S,C)
DO 7 J=1,NB1
7 F(J)=J-1
DO 500 J=1,NV-1
DO 500 K=J+1,NV
WRITE(0,300)CH(J),CH(K)
300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
CALL PLOT(C(1+NB1*(J-1)+NB1*NV1*(K-1)),F,NB)
100 FORMAT(1H ,F5.0,4X,E9.2/)
WRITE(0,102)
102 FORMAT(5(1H ,/))
500 CONTINUE
DO 105 J=1,NV
WRITE(0,102)
WRITE(0,107)CH(J)
107 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
CALL PLOT(C(1+NB1*(J-1)+NB1*NV1*(J-1)),F,NB)
105 CONTINUE
STOP
END
```

WHOISTEST

```
100 * DIMENSION X(NS*LX),R1(LR*NS*NS),W(M),F(M),S(M*NS*NS)
110 * DIMENSION C(M1*NS*NS),TLAG(2*LR-1),Z(2*LR-1),TLAGA(LR)
120 * NS=NUMBER OF CHANNELS
130 * LX=NUMBER OF DATA POINTS FROM EACH CHANNEL
140 * M1=M+1=LENGTH OF TIME LAG
150 * LR=MAXIMUN DESIRED TIME LAG .LE. LX
160 * L=SMALLEST INTEGER SUCH THAT LX<=2**L
170 * M=MAXIMUM LAG, M=2**(N-1)
180 * N IS DEFINED SO THAT 2*M=2**N
190 * H=LENGTH OF SAMPLING INTERVAL
200 *  LNXS=LX*NS
210 * LRNSNS=LR*NS*NS
220 * M1NSNS=M1*NS*NS
230 DIMENSION X(1000,2),R1(50,2,2),W(32),F(33),S(142),C(33,2,2),TLAG(99)
240 &,Z(99),TLAGA(50)
250 CHARACTER CH(2)/"1","2"/
260 DATA NS,LX,M,M1,LR,L,N,LXNS,LRNSNS,IN/2,1000,32,33,50,10,6,2000,200,2/
270 LIBRARY "REMAV","NLOGN","CROSS","WPARZ","MACOR","COQUAD","MOVE"
280 &,"NORMAG","COHERE","OLDPLO","GFSORT","BIG","SMALL","PLOTTR"
290 H=1./64.
300 OPENFILE 2,"WNSDAT","NUMERIC"
310 READ(2)X
315 CALL WPARZ(M,W)
320 DO 1 J=1,NS
330 1 CALL REMAV(LX,X(1,J))
340 CALL MACOR(NS,LX,X,LR,R1,LXNS,LRNSNS,L)
350 CALL COQUAD(H,NS,M,N,W,R1,S,M1,LR)
360 CALL COHERE(M1,NS,S,C)
370 DO 7 J=1,M1
380 7 F(J)=J-1
390 DO 500 J=1,NS-1
400 DO 500 K=J+1,NS
410 WRITE(0,300)CH(J),CH(K)
420 300 FORMAT(' COHERENCE FOR CHANNELS ',A1,' AND ',A1)
430 CALL OLDPLO(C(1,J,K),F,M)
450 100 FORMAT(1H ,2(F6.2,4X,E9.2,6X)/)
460 WRITE(0,102)
470 102 FORMAT(5(1H  ,/))
480 500 CONTINUE
490 DO 105 J=1,NS
500 WRITE(0,107)CH(J)
510 107 FORMAT(' AUTOSPECTRA FOR CHANNEL ',A1)
520 CALL OLDPLO(C(1,J,J),F,M)
540 105 WRITE(0,102)
550 DO 700 I=1,NS-1
560 DO 700 J=I+1,NS
570 WRITE(0,901)CH(I),CH(J)
580 901 FORMAT(' CROSS CORRELATION BETWEEN CHANNELS ',A1,' AND ',A1)
590 DO 108 L=1,LR-1
600 TLAG(L)=L-LR
610 108 Z(L)=R1(LR-L+1,J,I)
620 DO 109 L=LR,LR+LR-1
630 TLAG(L)=L-LR
640 109 Z(L)=R1(L-LR+1,I,J)
650 CALL PLOTTR(Z,TLAG,LR+LR-1)
660 WRITE(0,102)
670 700 CONTINUE
680 DO 701 K=1,LR
```

```
701 TLAGA(K)=K-1
DO 801 J=1,NS
WRITE(0,501)CH(J)
501 FORMAT(' AUTO CORRELATION FOR CHANNEL ',A1)
CALL PLOTTR(R1(1,J,J),TLAGA,LR)
801 WRITE(0,102)
PRINT,X
STOP
END
```

```
90 LIBRARY "FILTER","SPKFLT"
100 LIBRARY "SIMDAT","GENFLT","ORMFLT","WNOISE","FORTLIB***:FSAVFL"
101 &,"NLOGN","CROSS"
110 * LZ.GE.LX+2*L, L=50
120 * LX.GE.LY+2*NW, NW=50
130 DIMENSION Z(1000),X(1000,4),XF(8),YF(8),Y(800)
140 &,AW(101),Q(9),WT(1000),XZ(1000,5),C(4)
145 &,A(4,5),D(4,5),W(4,5)
150 DATA H,LZ,LX,LY,NWT,LN2N/0.015625,1000,900,800,101,10/
160 DATA C(1),C(2),C(3),C(4)/0.4,0.1,0.1,0.1/
190 DATA XF/4.,5.,7.,8.,12.,13.,15.,16./
200 DATA YF/0.,1.,1.,0.,0.,0.5,0.5,0./
210 DATA A(1,1),D(1,1),W(1,1)/2.,0.,1./
220 DATA A(1,2),D(1,2),W(1,2)/4.,6.,1./
230 DATA A(1,3),D(1,3),W(1,3)/3.,10.,1./
240 DATA A(1,4),D(1,4),W(1,4)/4.5,14.,2./
250 DATA A(1,5),D(1,5),W(1,5)/4.,29.,1./
260 DATA A(2,1),D(2,1),W(2,1)/.1,10.,1./
270 DATA A(2,2),D(2,2),W(2,2)/.7,20.,1./
280 DATA A(3,1),D(3,1),W(3,1)/.8,3.,1./
290 DATA A(3,2),D(3,2),W(3,2)/.2,25.,1./
300 DATA A(4,1),D(4,1),W(4,1)/.25,12.,1./
310 DATA A(4,2),D(4,2),W(4,2)/0.,1.,1./
320 DO 1 K=1,5
340 1 CALL SIMDAT(H,A(1,K),D(1,K),W(1,K),LZ,Z,LX,XZ(1,K),NWT,AW,WT,LN2N)
345 CALL WNOISE(LZ,Z,C(1))
350 DO 2 J=1,LX
360 X(J,1)=Z(J)
370 DO 2 K=1,5
380 2 X(J,1)=X(J,1)+XZ(J,K)
390 CALL ORMFLT(LY,Y,NWT,AW,LZ,X(1,1),H,XF,YF,8,9,Q,WT,LN2N)
400 DO 3 N=2,4
410 DO 4 K=1,2
420 4 CALL SIMDAT(H,A(N,K),D(N,K),W(N,K),LZ,Z,LY,XZ(1,K),NWT,AW,WT,LN2N)
430 CALL WNOISE(LZ,Z,C(N))
440 DO 3 J=1,LY
450 X(J,N)=0.
460 DO 5 K=1,2
470 5 X(J,N)=X(J,N)+XZ(J,K)
480 3 X(J,N)=X(J,N)+Z(J)+Y(J)
490 CALL FSAVFL("FCHDAT"," ",I)
500 OPENFILE 2,"FCHDAT","NUMERIC"
510 WRITE(2)((X(I,J),I=1,LY),J=1,4)
520 STOP
530 END
```

```
SUBROUTINE ORMFLT(LY,Y,NWT,AW,LX,X,H,XF,YF,LW,LW2,Q,WT,LN2N)
DIMENSION Y(LY),X(LX),AW(NWT),WT(LX),XF(LW),YF(LW),Q(LW2)
* LX.GE.LY+2*NW
* LW2.GE.LW+1
NW=(NWT-1)/2
NW1=NW+1
CALL GENFLT(H,AW(NW1),NW1,XF,YF,LW,Q,LW2)
DO 1 J=1,NW
1 AW(NW1-J)=AW(NW1+J)
CALL FILTER(LX,NWT,AW,WT,X,LY,Y,LN2N)
 RETURN
END
```

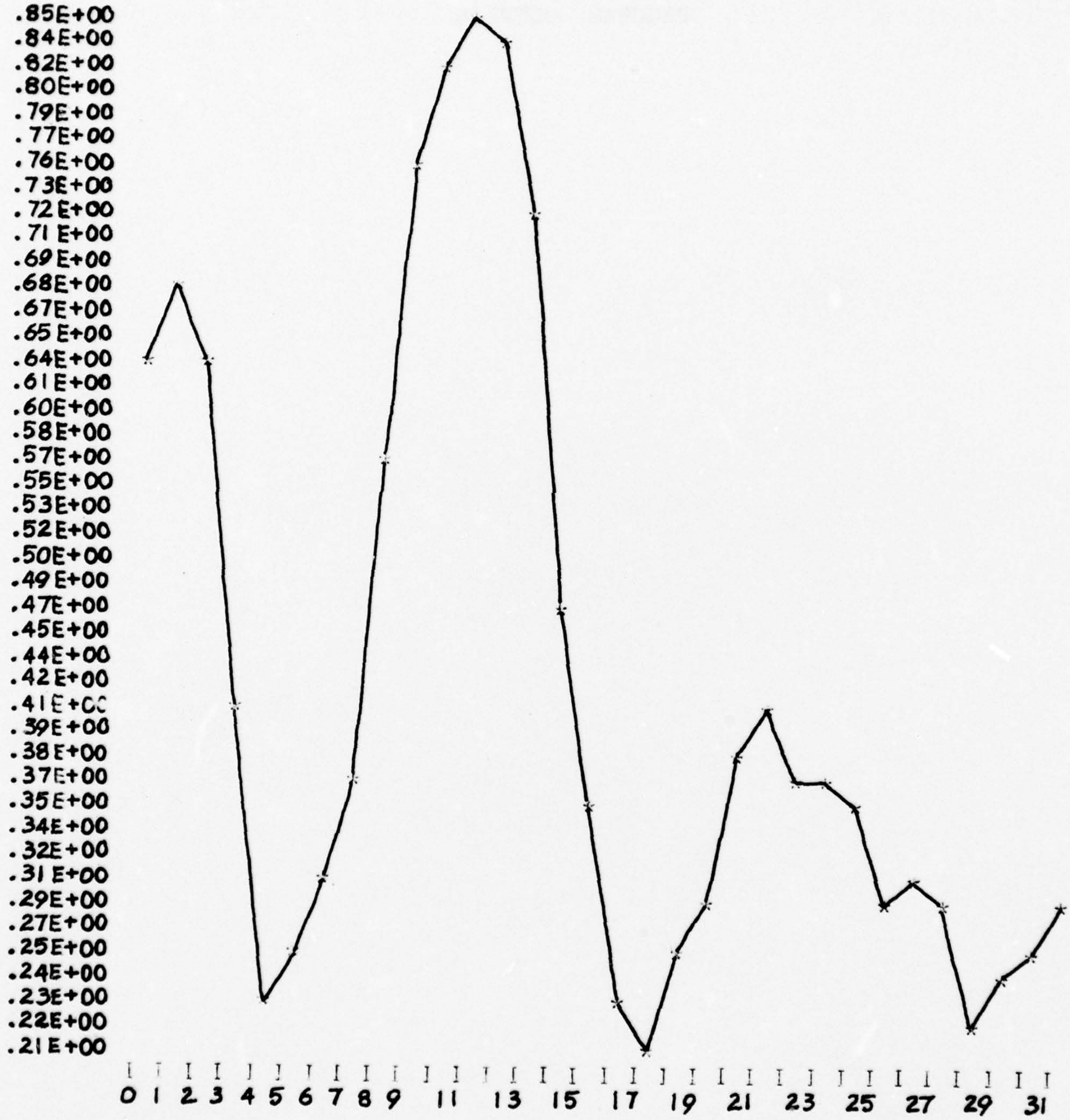
```
SUBROUTINE SPKFLT(H,A1,D,W,NW1,A)
DIMENSION A(NW1)
P=6.2831853
C=A1/(19.73925*H*W)
IF(D.EQ.0.0)GO TO 200
A(1)=(P*H*W)**2
C1=P*H*(D-W)
C2=P*H*D
C3=P*H*(D+W)
DO 1 J=2,NW1
K=J-1
A(J)=-(1./K**2)*(COS(C1*K)-2.*COS(C2*K)+COS(C3*K))
1 CONTINUE
DO 2 J=1,NW1
2 A(J)=C*A(J)
RETURN
200 C1=P*H*W
A(1)=(0.5)*C1**2
DO 3 J=2,NW1
K=J-1
3 A(J)=(-1.0)*(1./K**2)*(COS(C1*K)-1.0)
DO 4 J=1,NW1
4 A(J)=C*A(J)
RETURN
END
```

```
SUBROUTINE SIMDAT(H,A1,D,W,LZ,Z,LX,X,NWT,AW,WT,LN2N)
DIMENSION Z(LZ),X(LX),AW(NWT),WT(LZ)
* LZ.GE.LX+2*NW=LX+NWT-1
NW=(NWT-1)/2
NW1=NW+1
CALL SPKFLT(H,A1,D,W,NW1,AW(NW1))
DO 1 J=1,NW
1 AW(NW1-J)=AW(NW1+J)
CALL WNOISE(LZ,Z,1.0)
CALL FILTER(LZ,NWT,AW,WT,Z,LX,X,LN2N)
RETURN
END
```

APPENDIX B

PROGRAM OUTPUTS

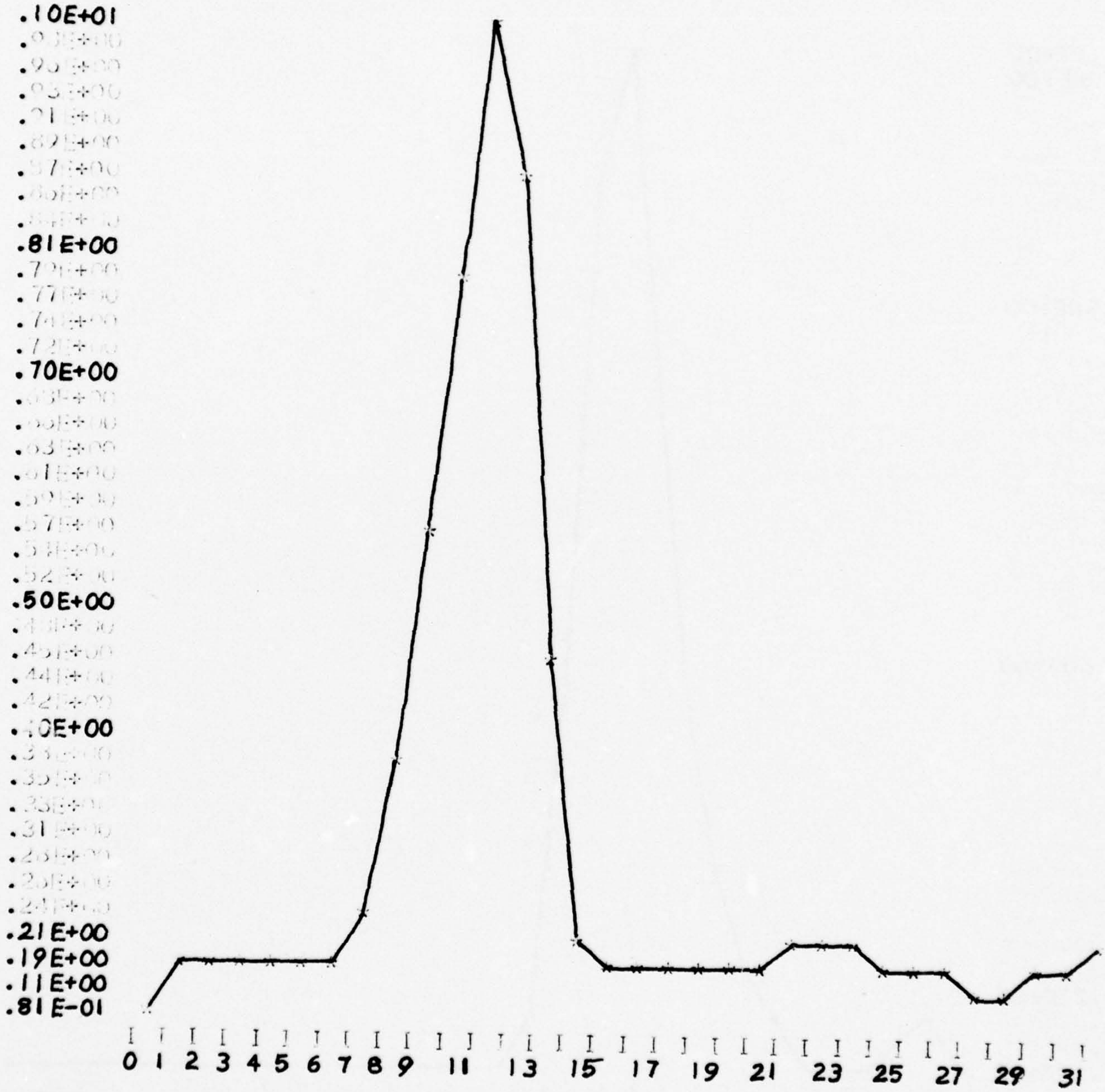
COHERENCE FOR CHANNELS 1 AND 2



EEG COHERENCE ESTIMATE OBTAINED BY USING PROGRAM MULSPECT

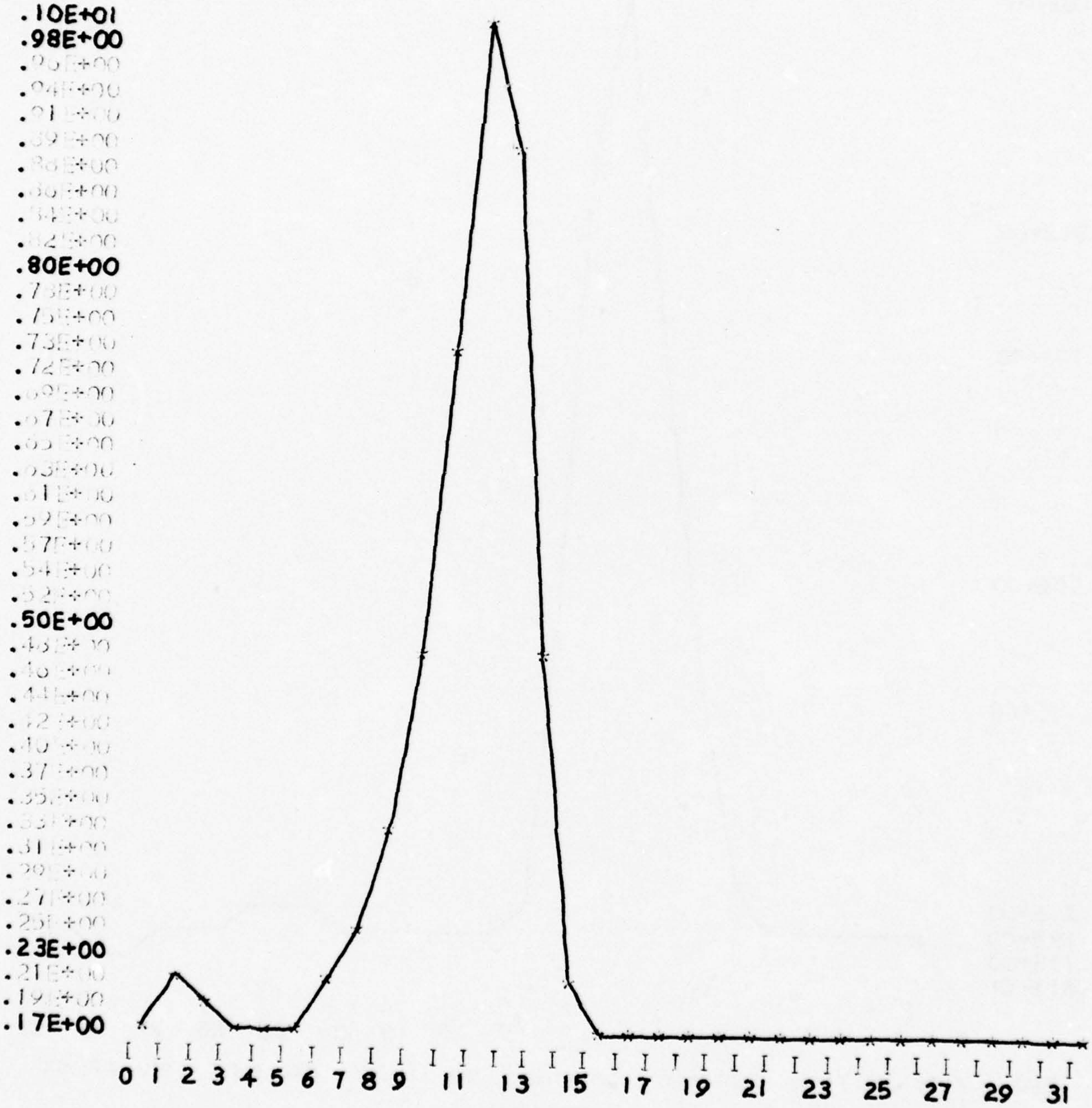
SAMPLING RATE: 64 PER SECOND, DEGREES OF FREEDOM: 37

AUTOSPECTRA FOR CHANNEL 1



EEG AUTOSPECTRUM ESTIMATE OBTAINED BY USING PROGRAM MULSPECT
37 DEGREES OF FREEDOM

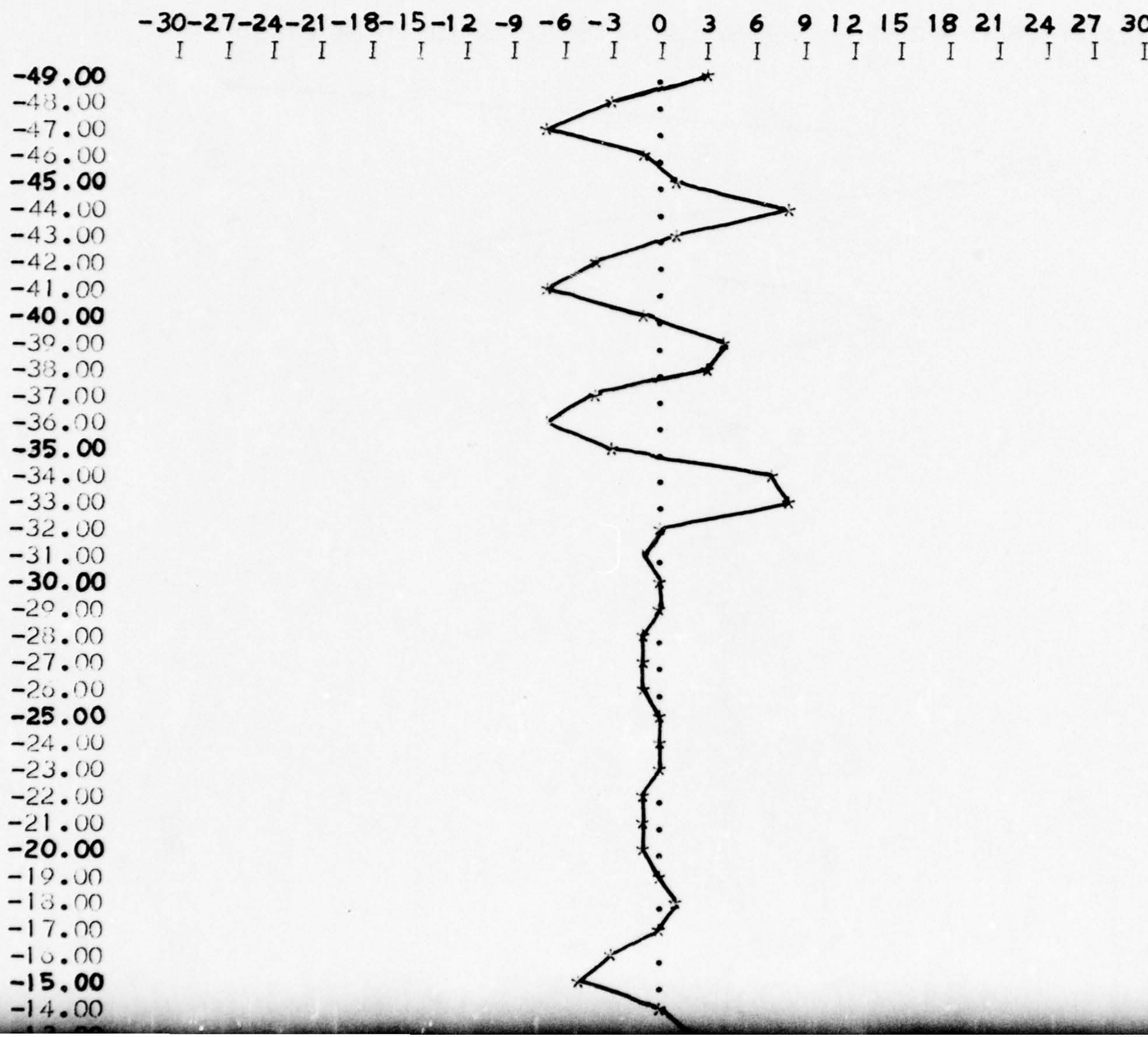
AUTOSPECTRA FOR CHANNEL 2

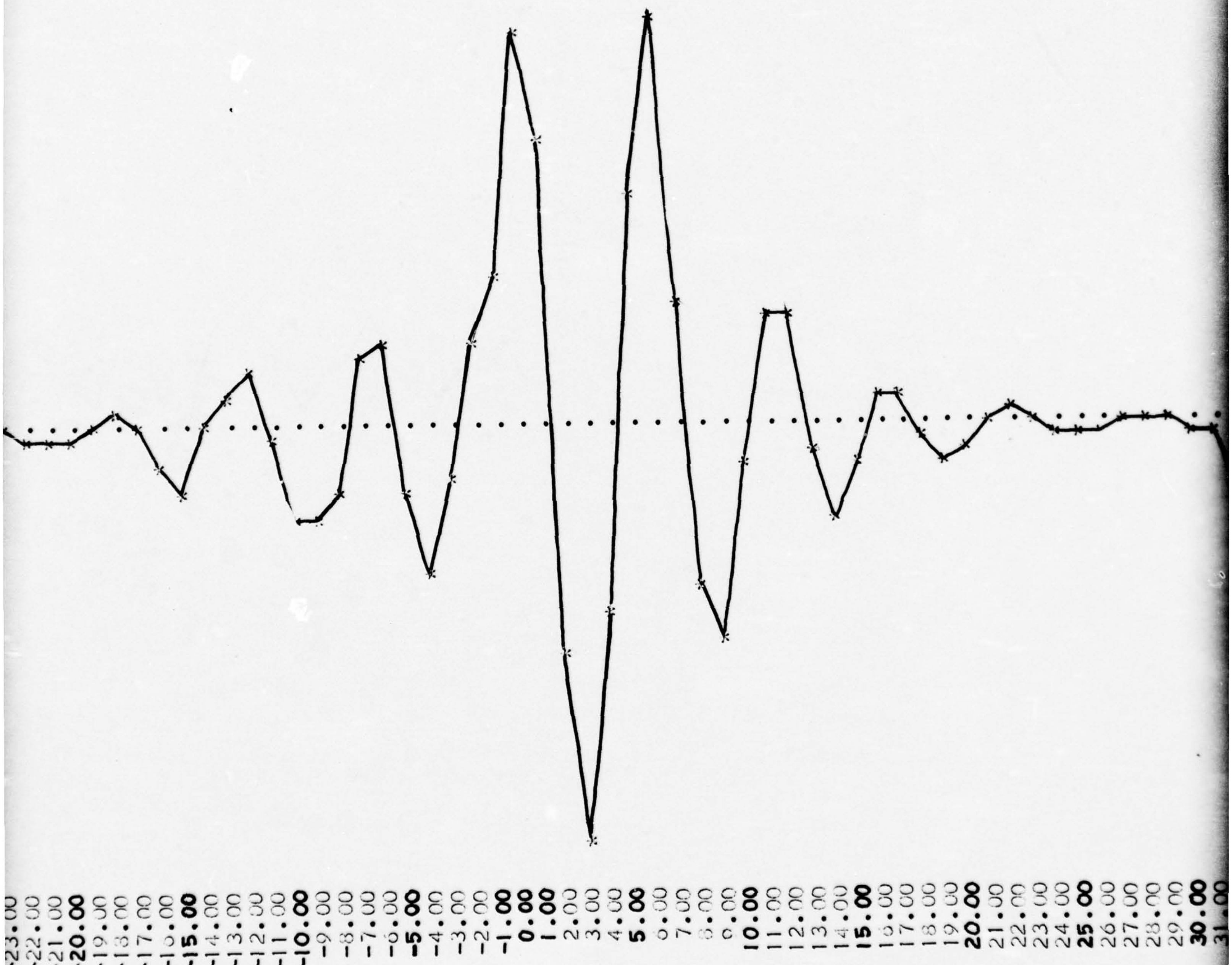


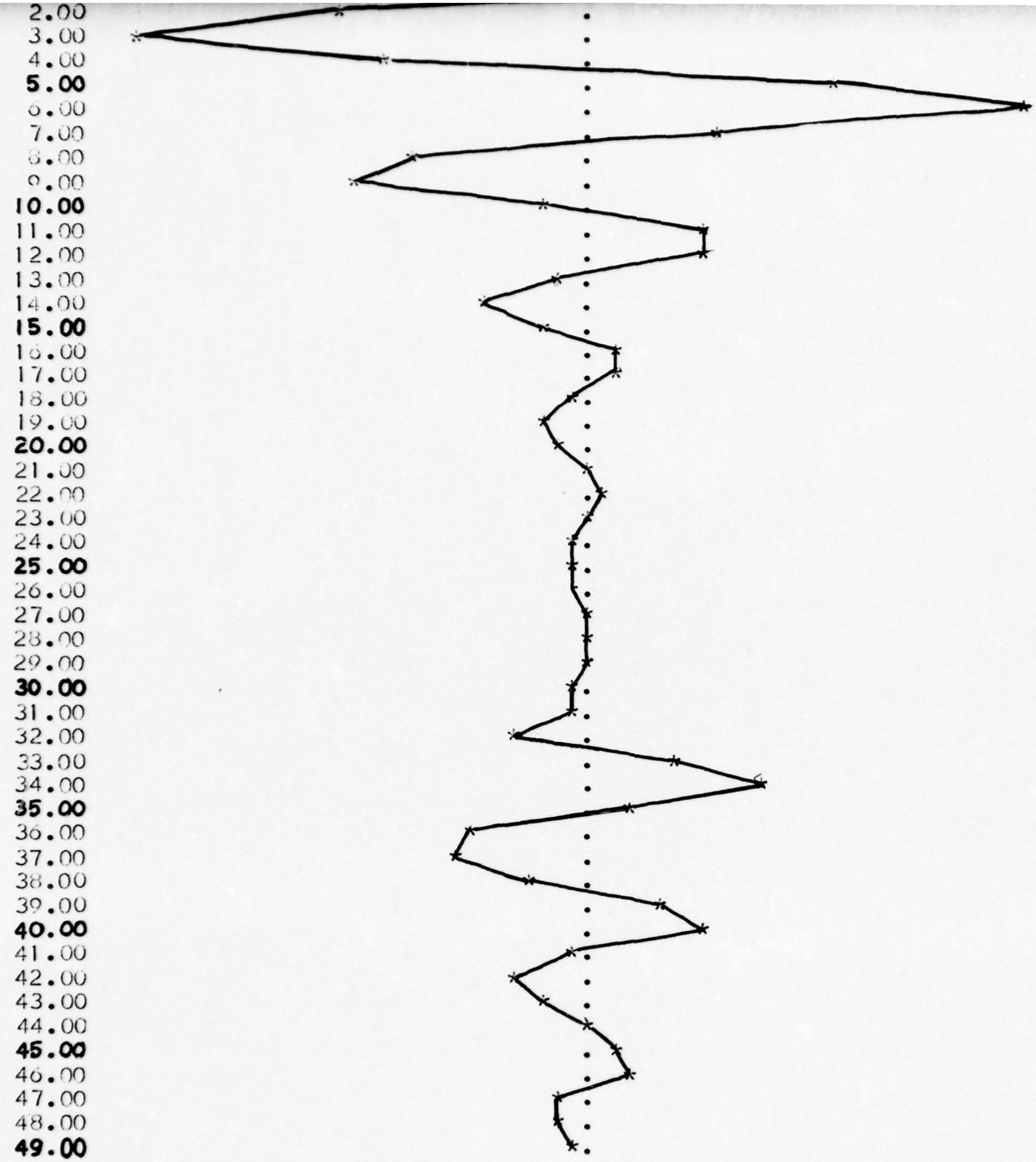
EEG AUTOSPECTRUM ESTIMATE OBTAINED BY USING PROGRAM MULSPECT
37 DEGREES OF FREEDOM

CROSS CORRELATION BETWEEN CHANNELS 1 AND 2
LARGEST VALUE IS .1401775E+07
SMALLEST VALUE IS -.1426841E+07
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .4602714E+05

113



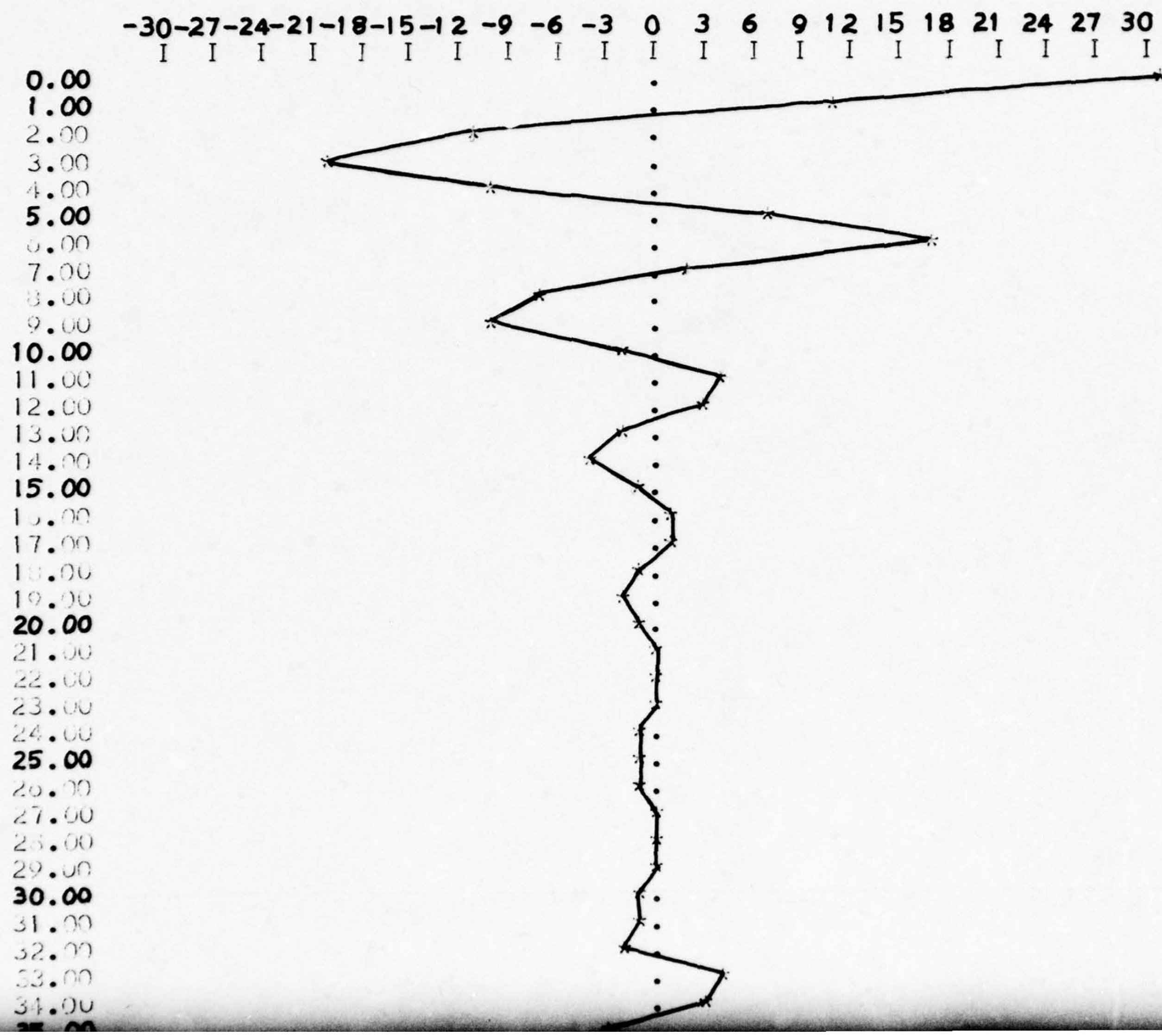

-23.00
-22.00
-21.00
-20.00
-19.00
-18.00
-17.00
-16.00
-15.00
-14.00
-13.00
-12.00
-11.00
-10.00
-9.00
-8.00
-7.00
-6.00
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00
17.00
18.00
19.00
20.00
21.00
22.00
23.00
24.00
25.00
26.00
27.00
28.00
29.00
30.00
31.00

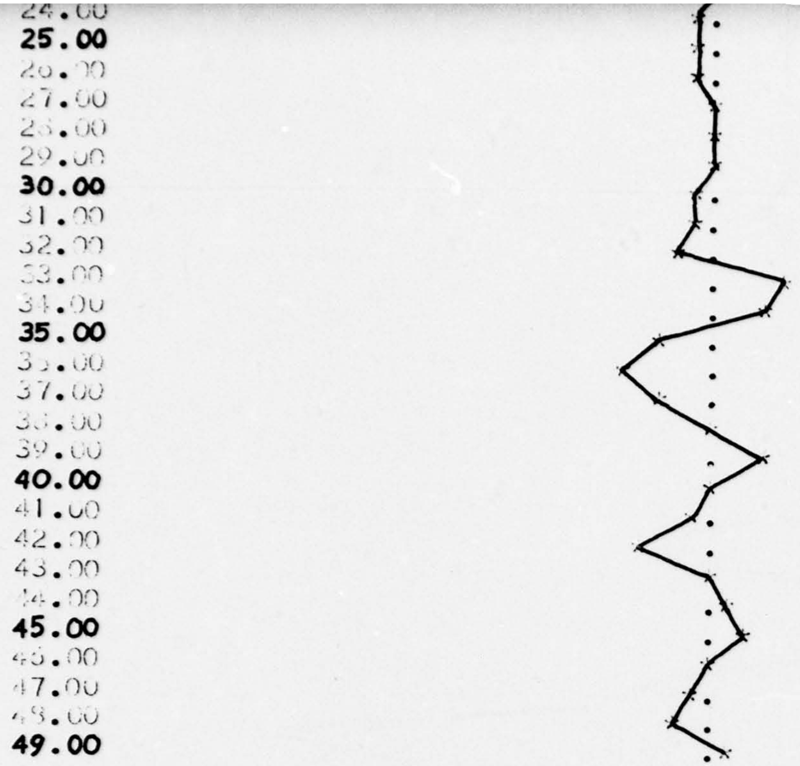


EEG CROSS CORRELATION FUNCTION OBTAINED BY USING PROGRAM MULSPECT

3

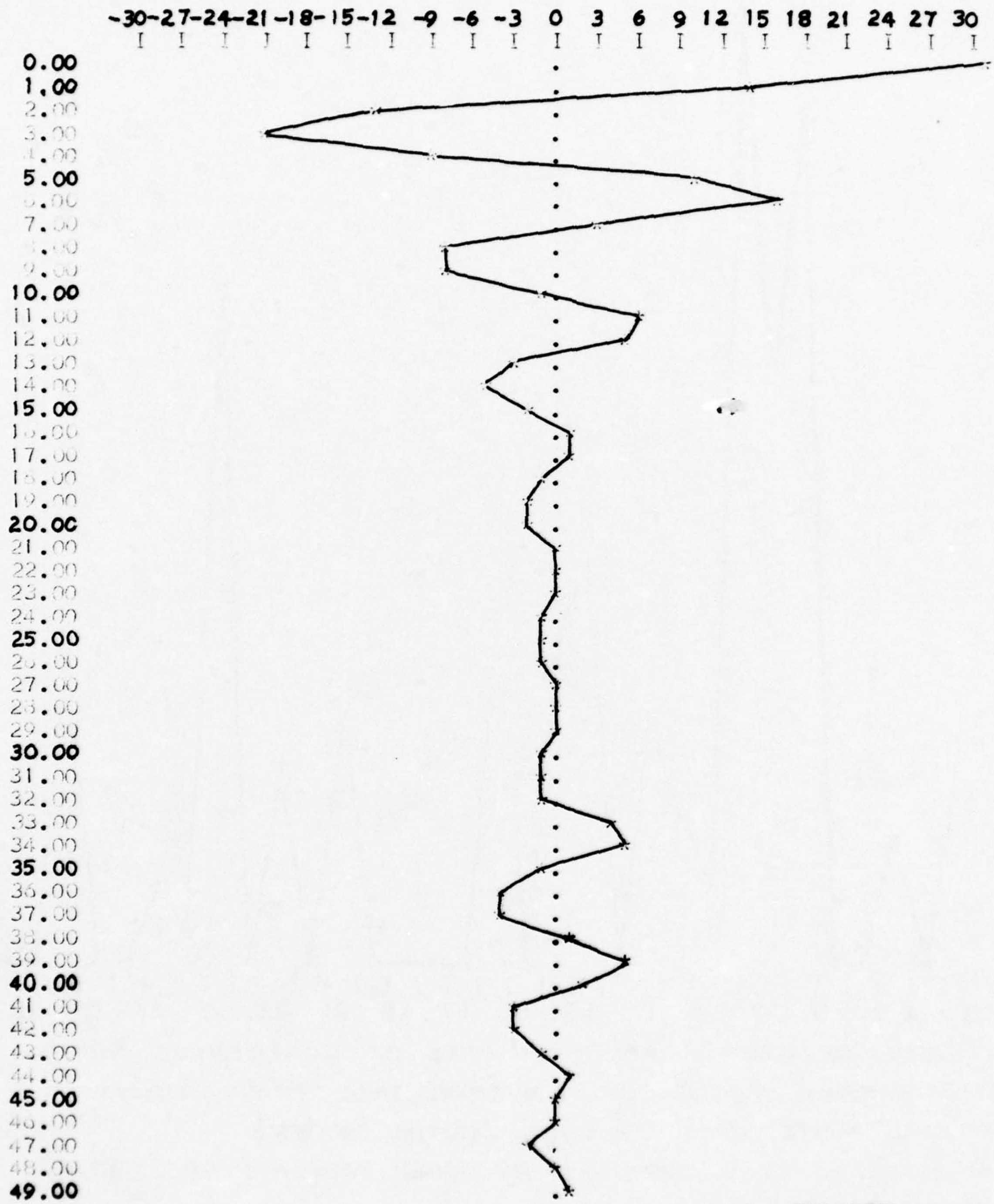
AUTO CORRELATION FOR CHANNEL 1
LARGEST VALUE IS .2638448E+07
SMALLEST VALUE IS -.1674095E+07
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .8511122E+05





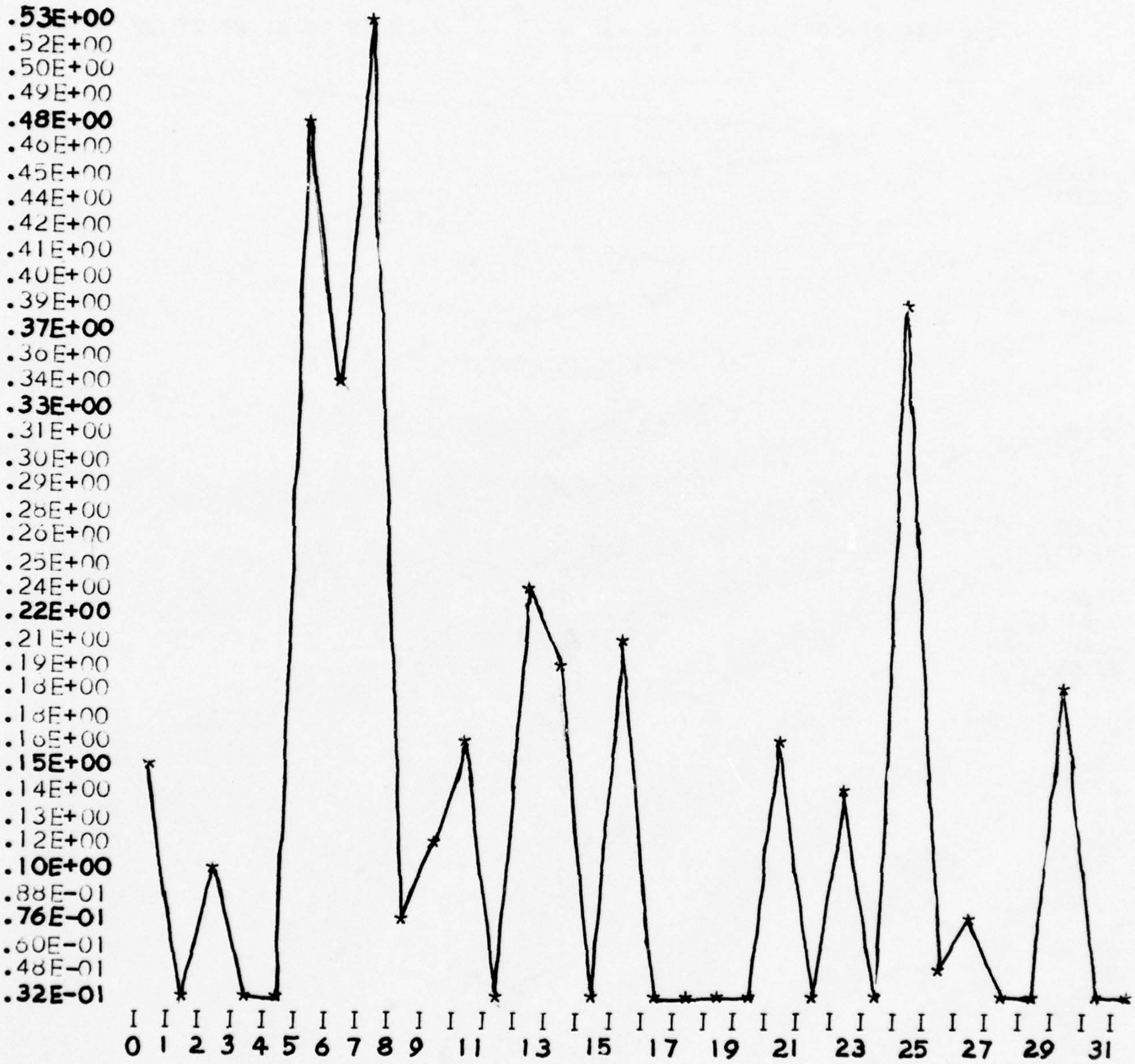
EEG AUTO CORRELATION FUNCTION OBTAINED BY USING PROGRAM MULSPECT

AUTO CORRELATION FOR CHANNEL 2
LARGEST VALUE IS .2506558E+07
SMALLEST VALUE IS -.1673505E+07
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .8085670E+05



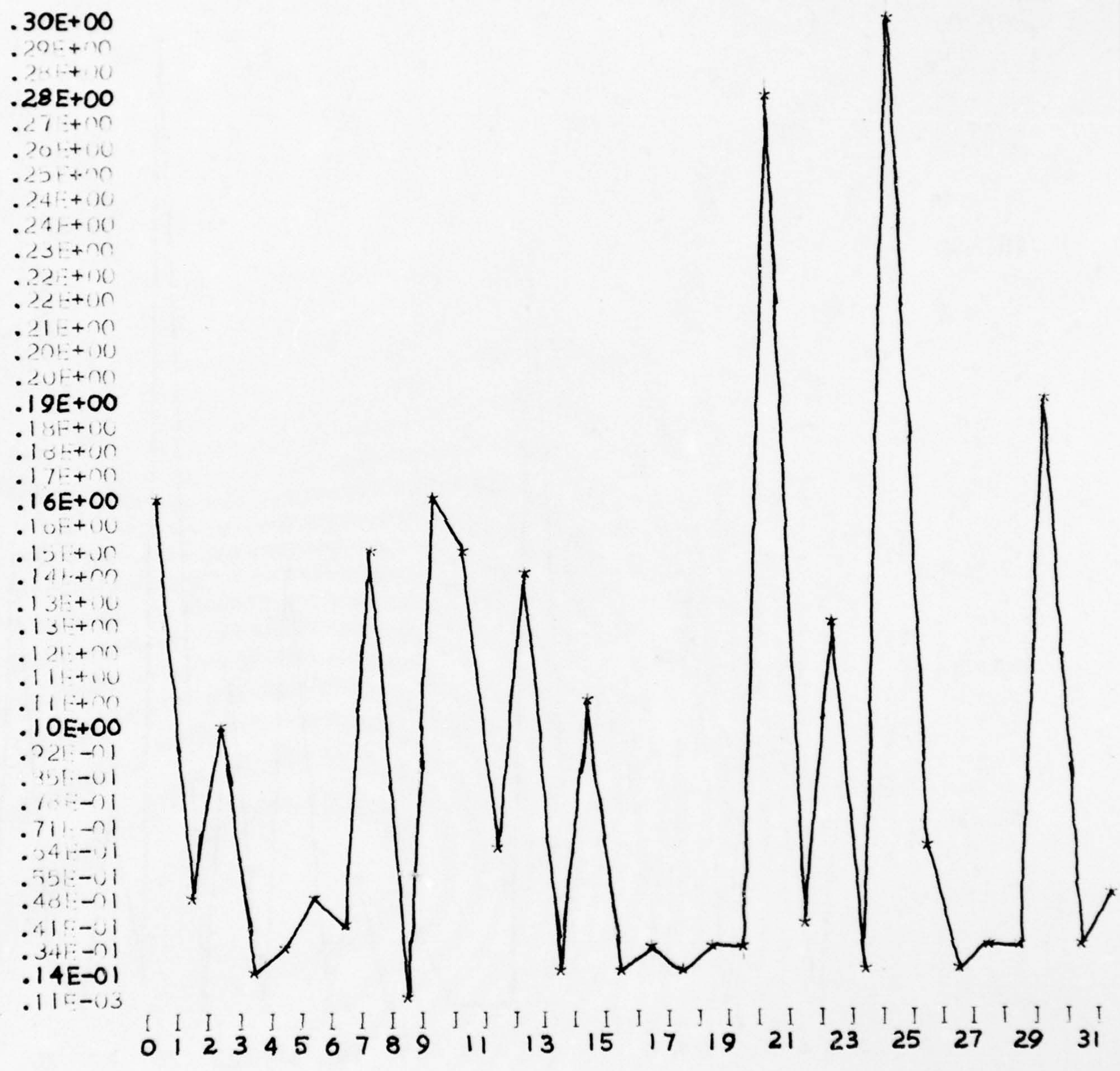
EEG AUTOCORRELATION FUNCTION OBTAINED BY USING PROGRAM MULSPECT

COHERENCE FOR CHANNELS 1 AND 2

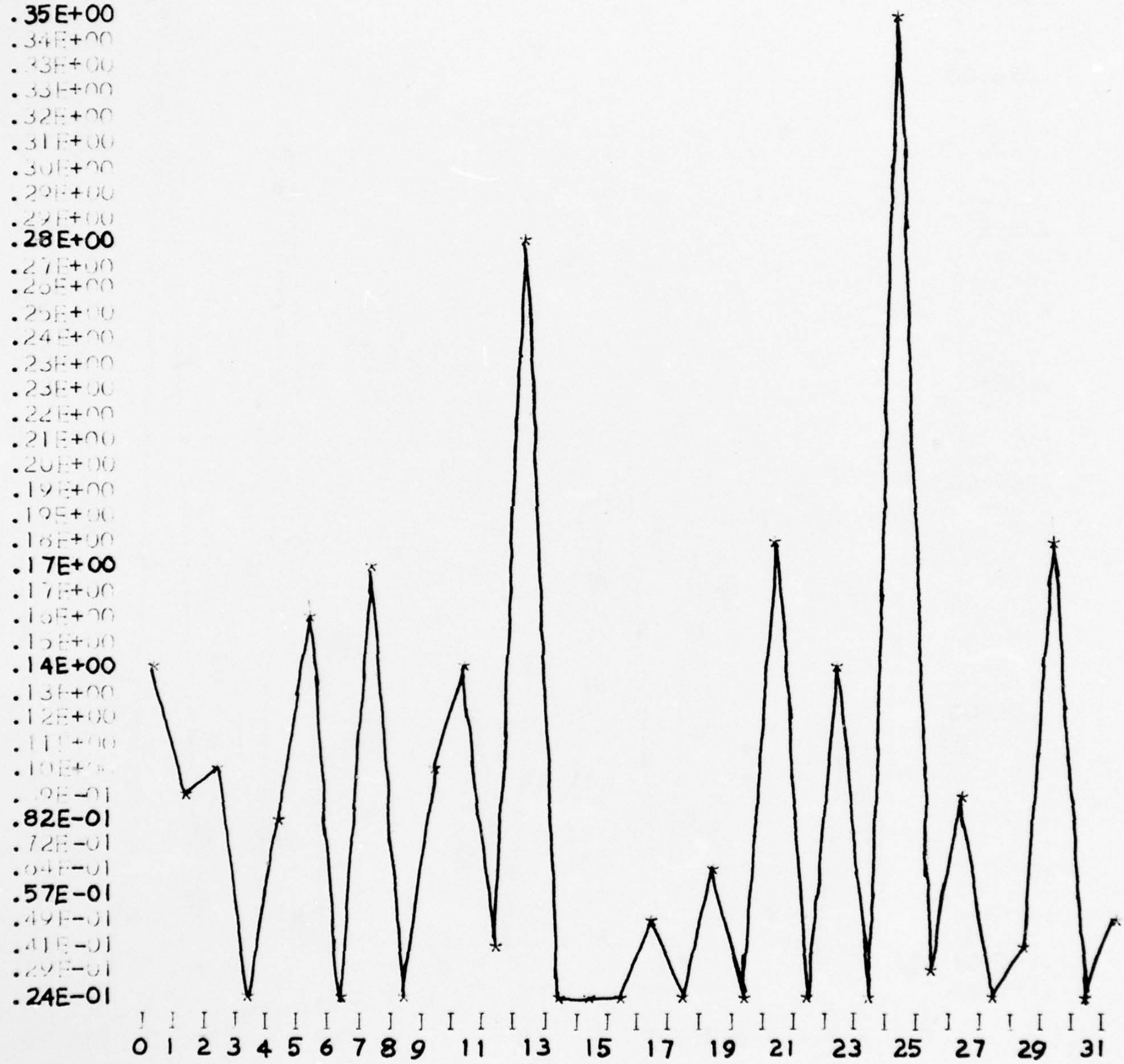


THE NEXT 22 GRAPHS ARE ESTIMATES OF COHERENCES AND PARTIAL COHERENCE SPECTRA OF 4 CHANNEL TIME SERIES, OBTAINED BY PASSING WHITE NOISE THROUGH DIGITAL FILTERS.
THE ESTIMATES WERE OBTAINED BY USING PROGRAM SPCTBGTK, WITH 25 DEGREES OF FREEDOM.

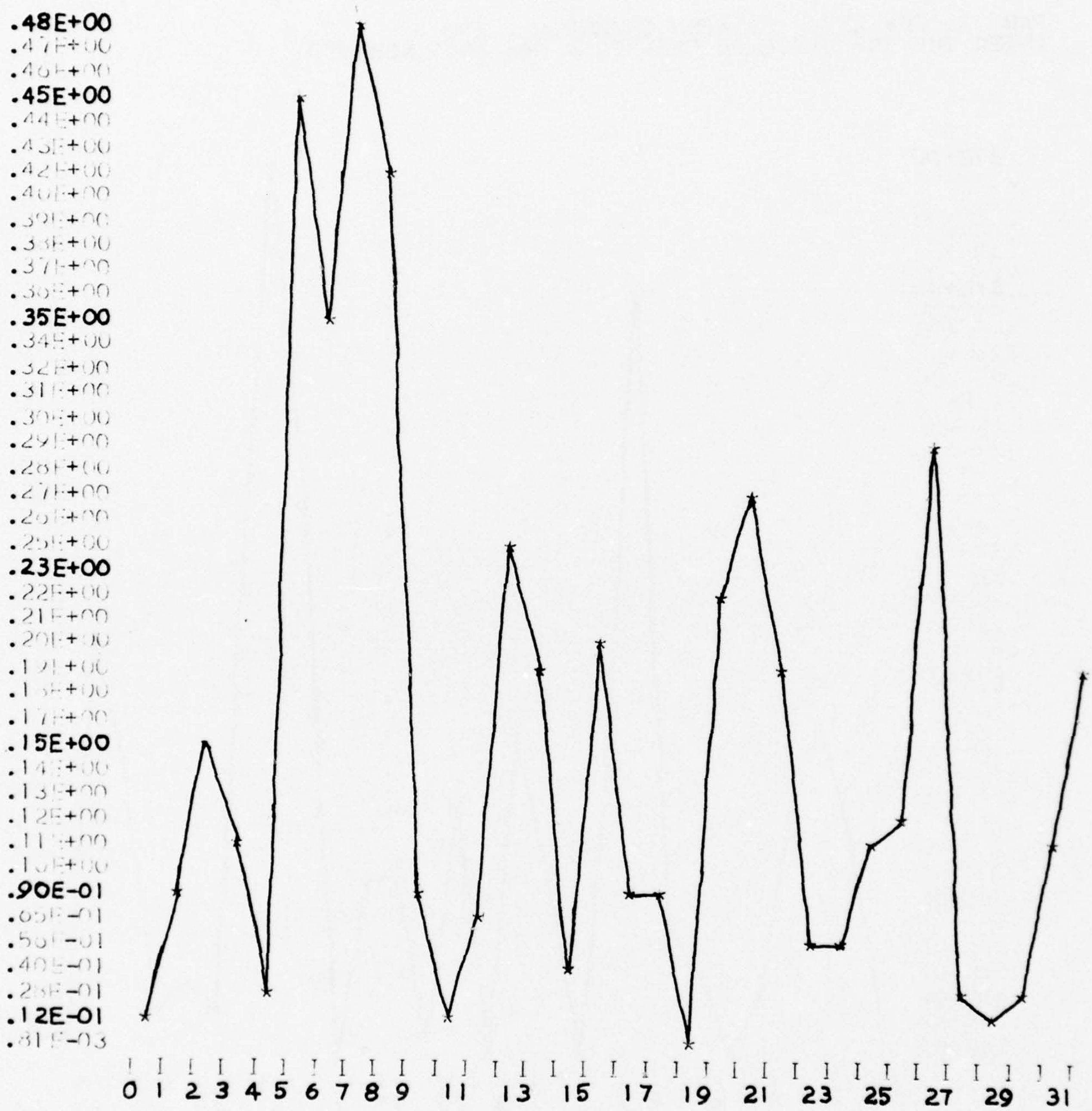
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 2
AFTER THE INFLUENCE OF CHANNEL 3 HAS BEEN REMOVED



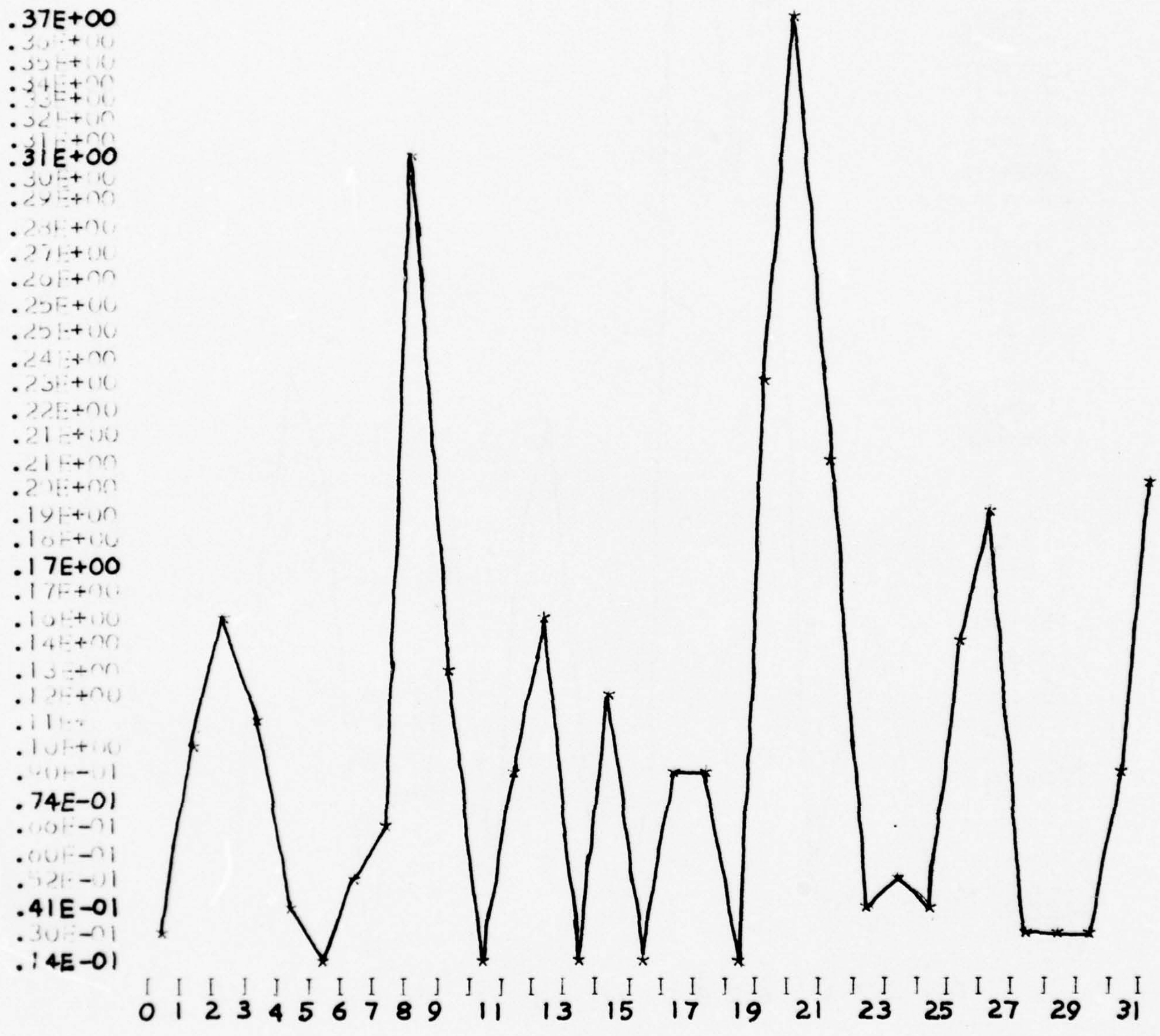
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 2
AFTER THE INFLUENCE OF CHANNEL 4 HAS BEEN REMOVED



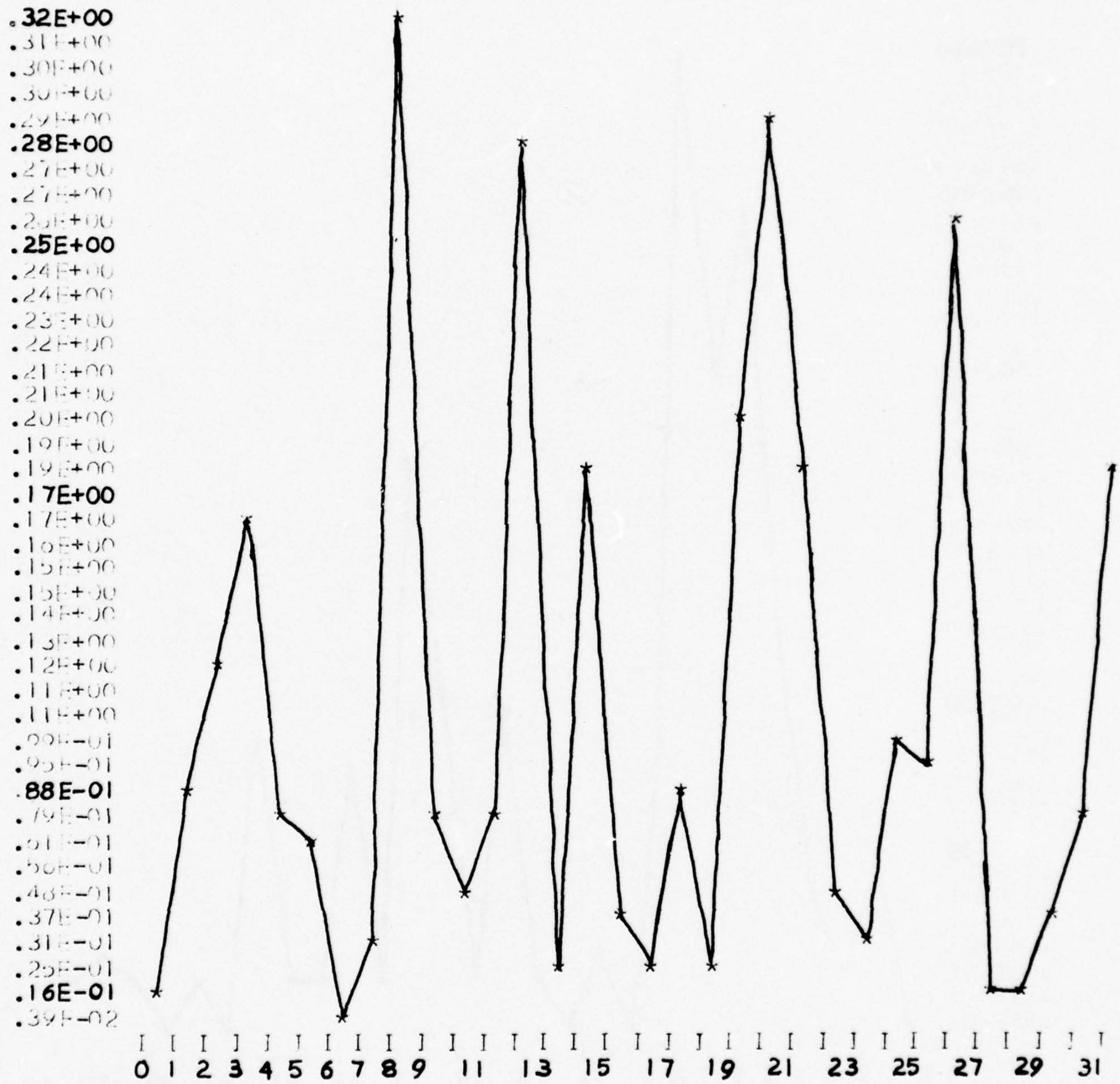
COHERENCE FOR CHANNELS 1 AND 3



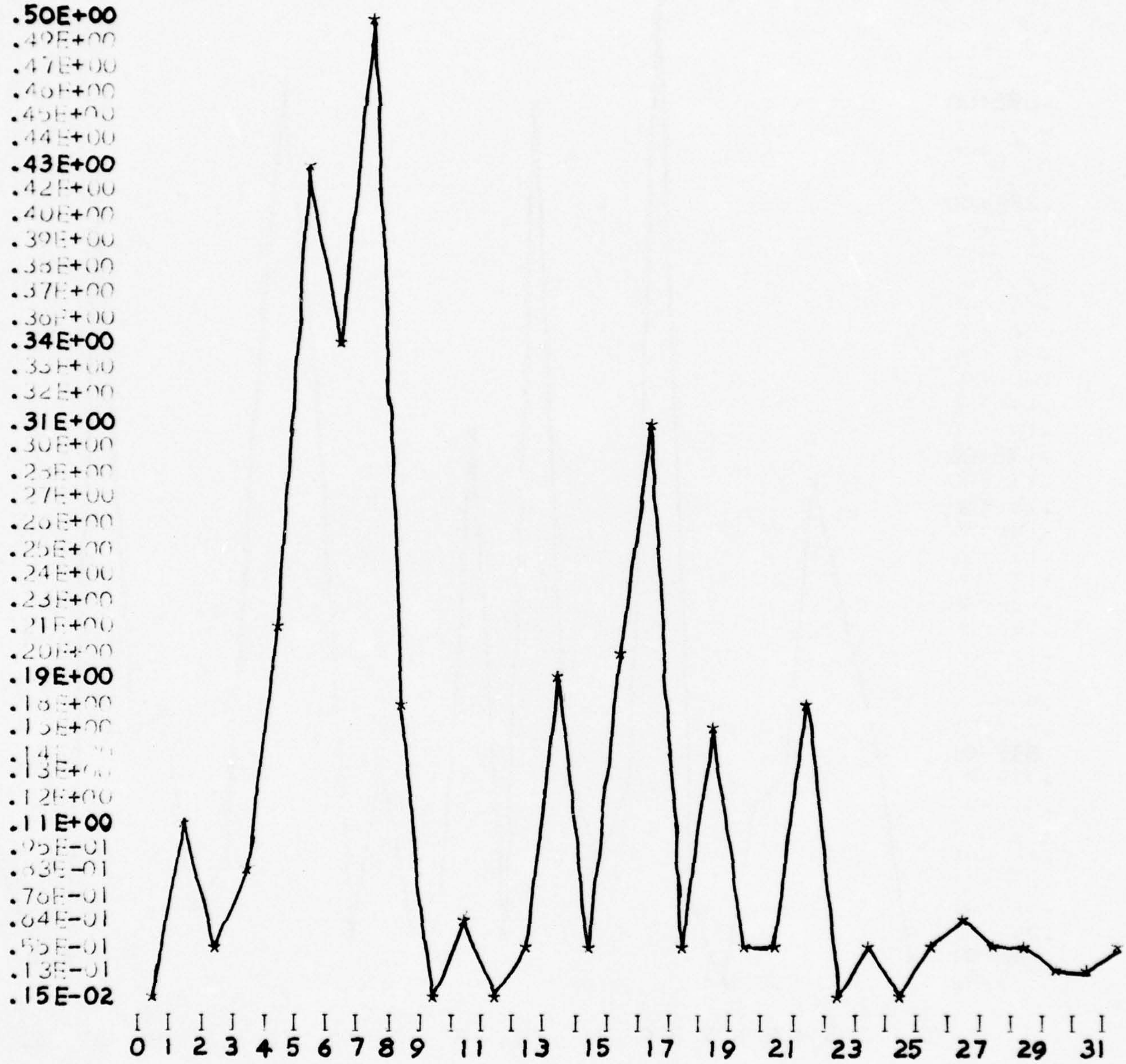
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 3
AFTER THE INFLUENCE OF CHANNEL 2 HAS BEEN REMOVED



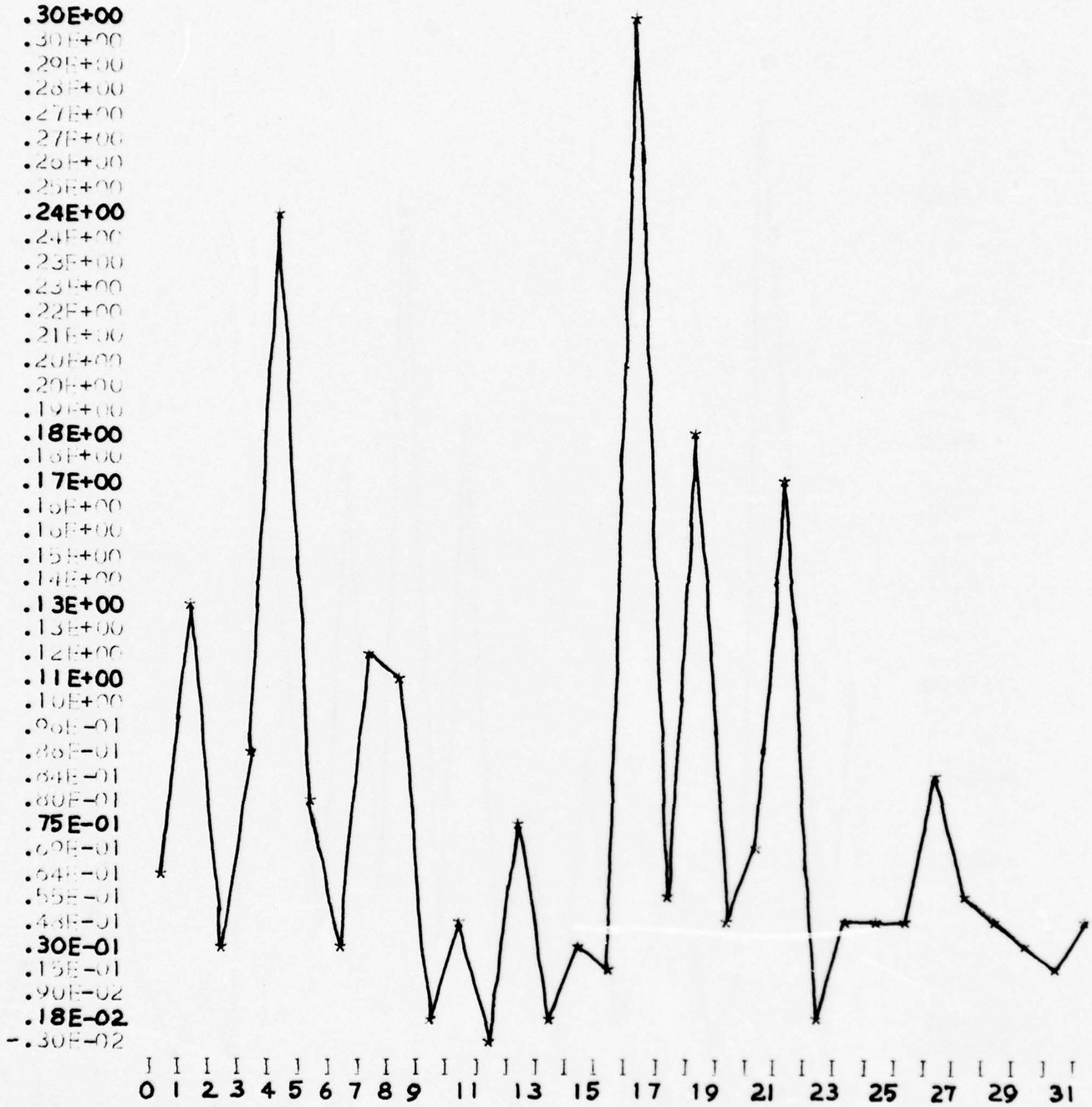
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 3
AFTER THE INFLUENCE OF CHANNEL 4 HAS BEEN REMOVED



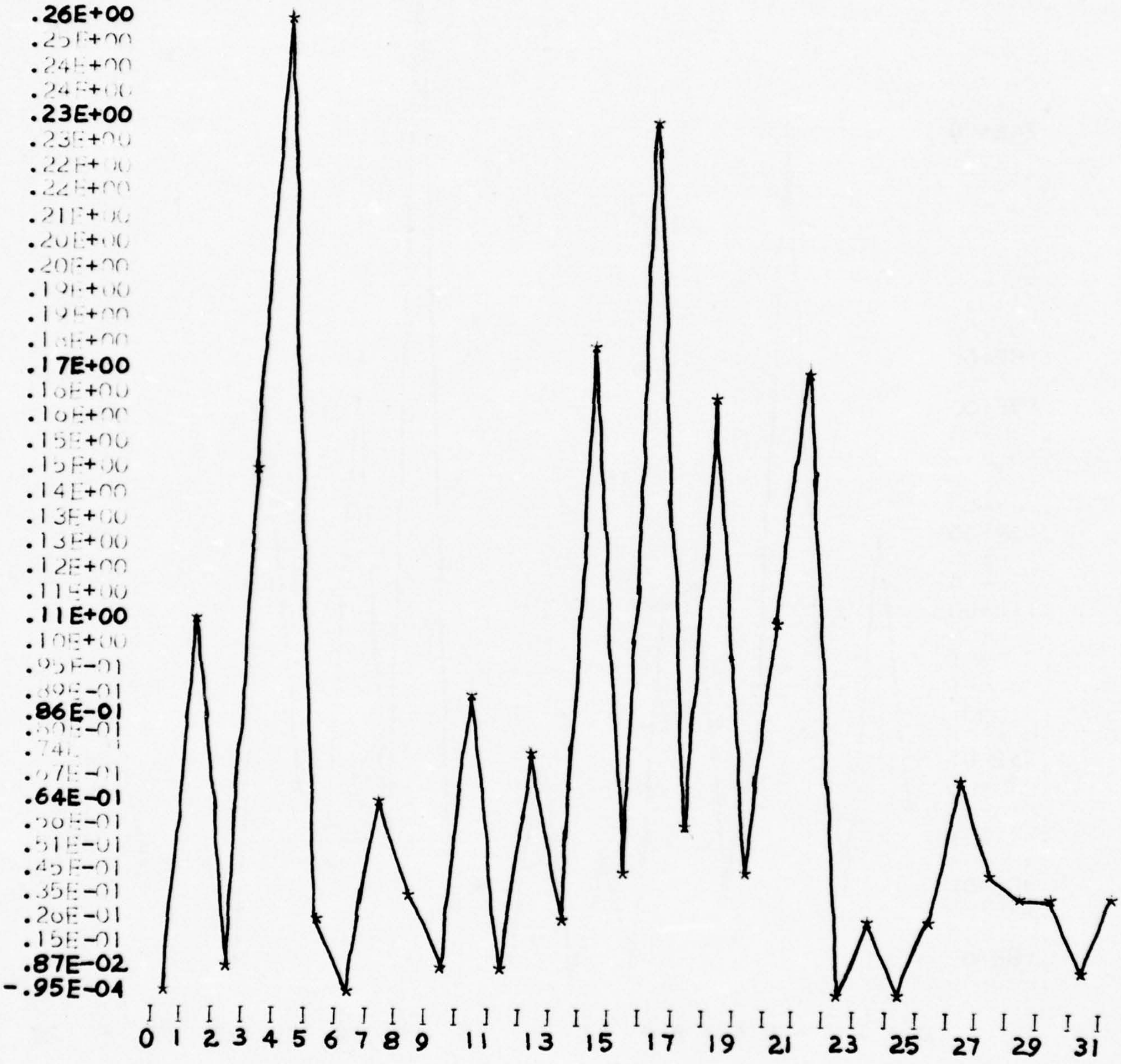
COHERENCE FOR CHANNELS 1 AND 4



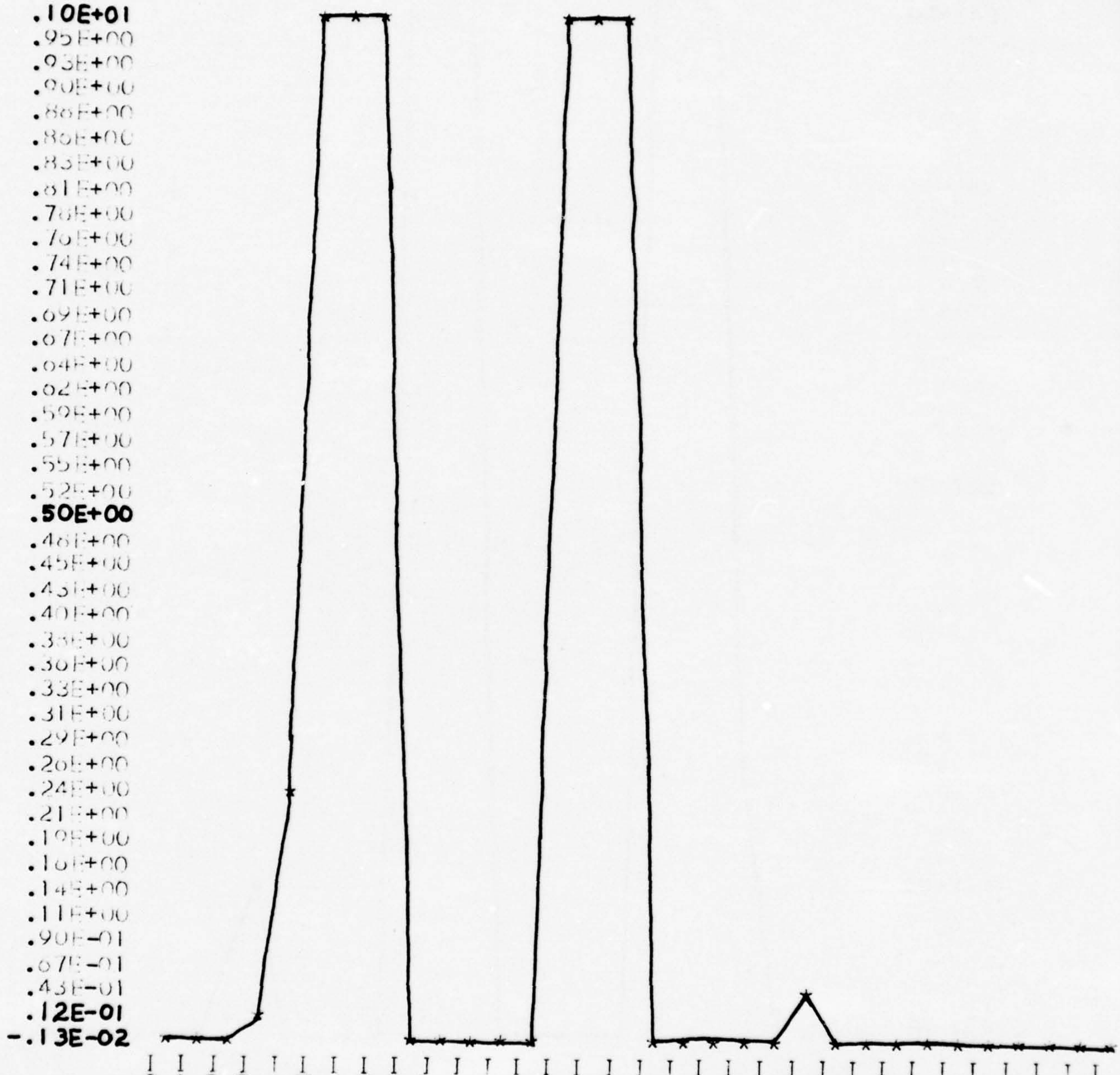
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 4
AFTER THE INFLUENCE OF CHANNEL 2 HAS BEEN REMOVED



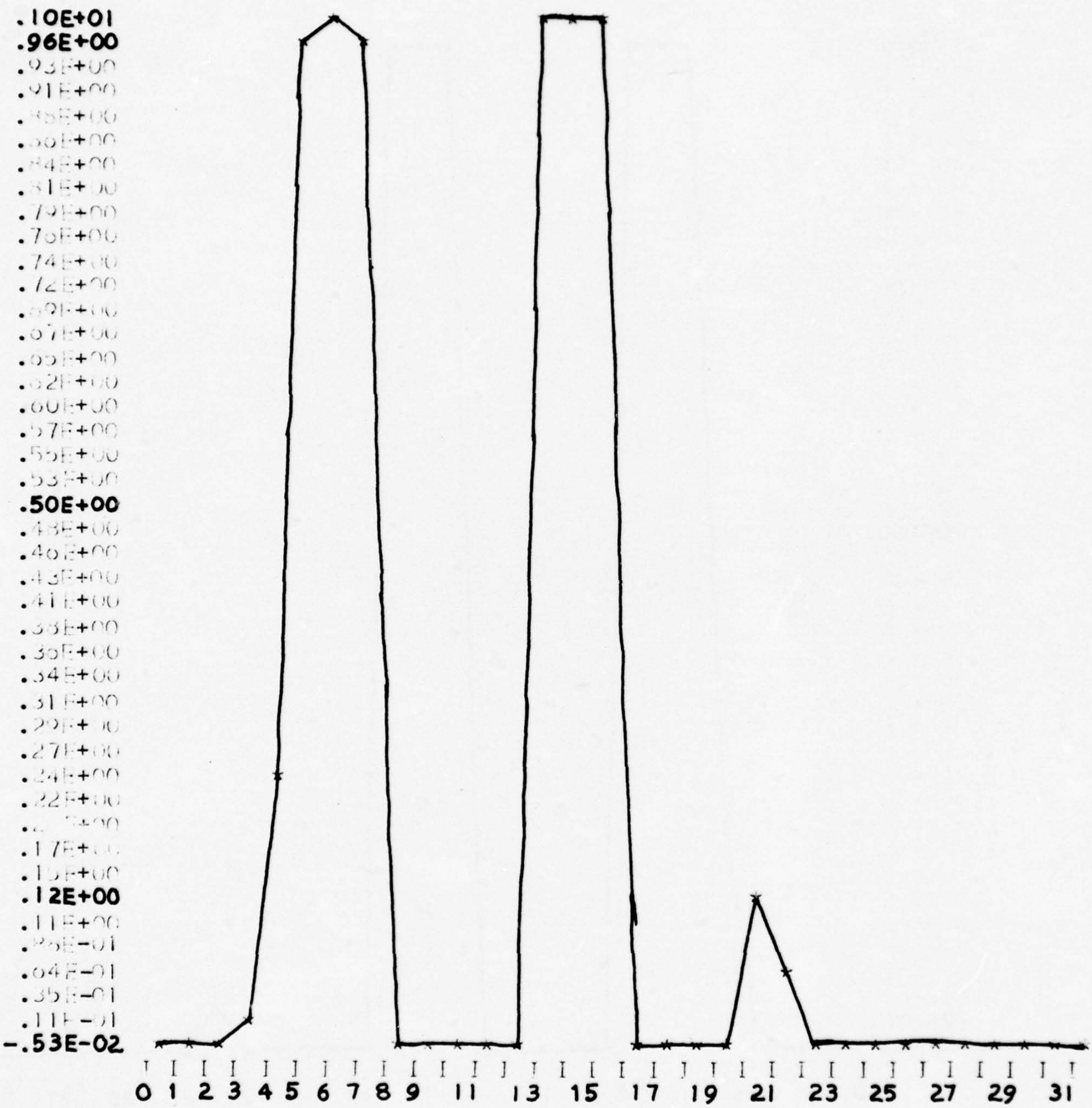
PARTIAL COHERENCE BETWEEN CHANNELS 1 AND 4
AFTER THE INFLUENCE OF CHANNEL 3 HAS BEEN REMOVED



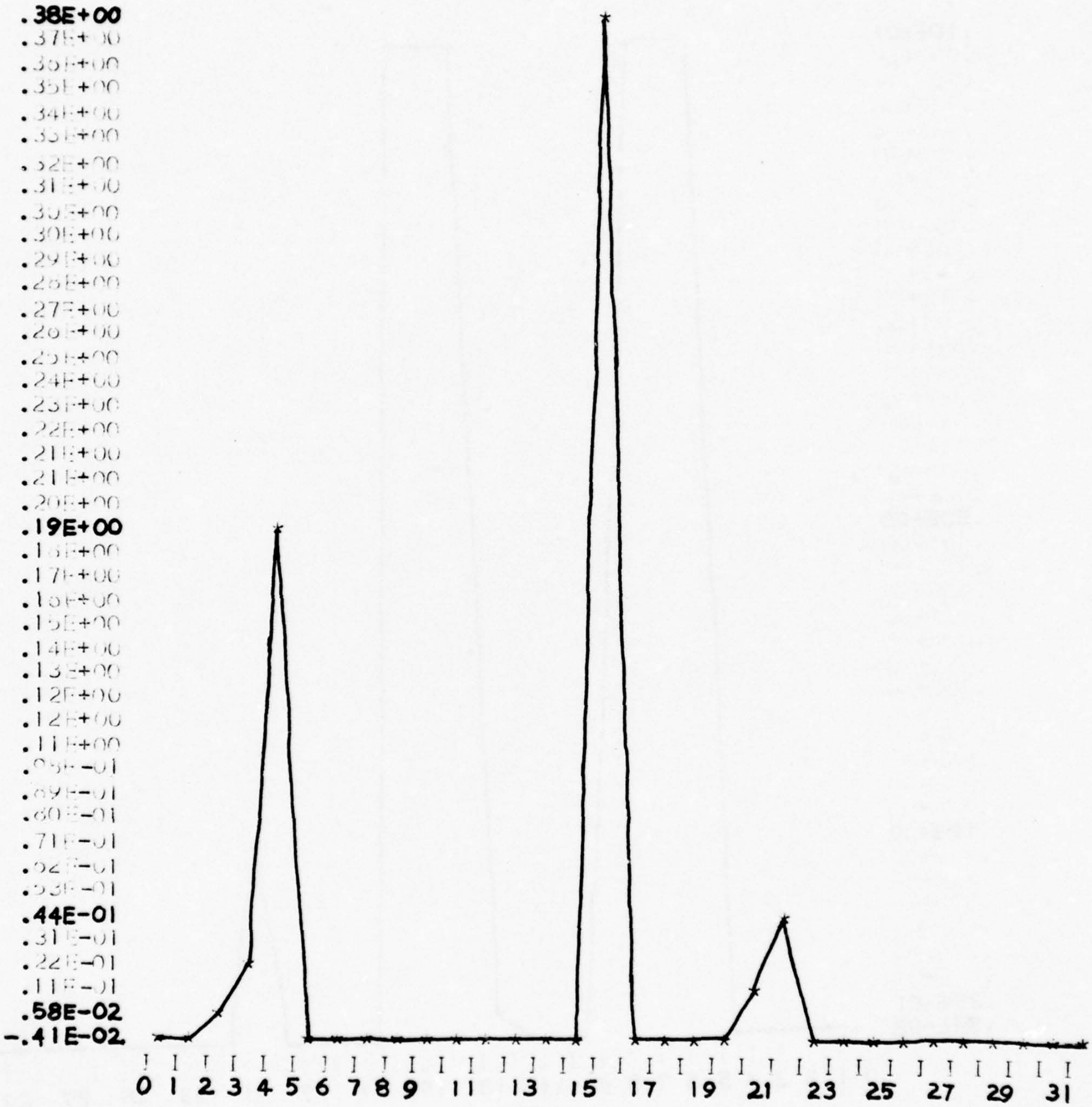
COHERENCE FOR CHANNELS 2 AND 3



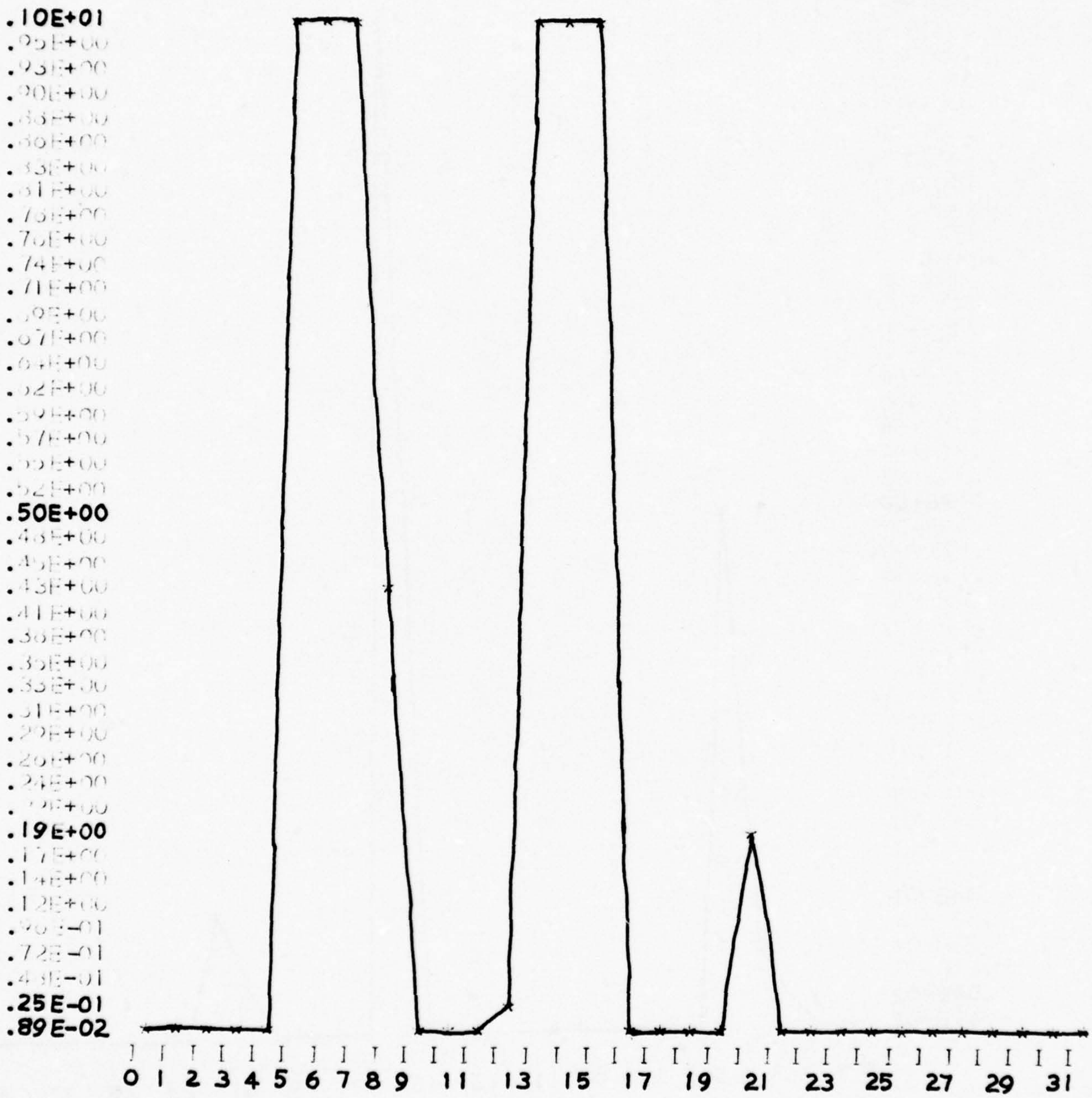
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 3
AFTER THE INFLUENCE OF CHANNEL 1 HAS BEEN REMOVED



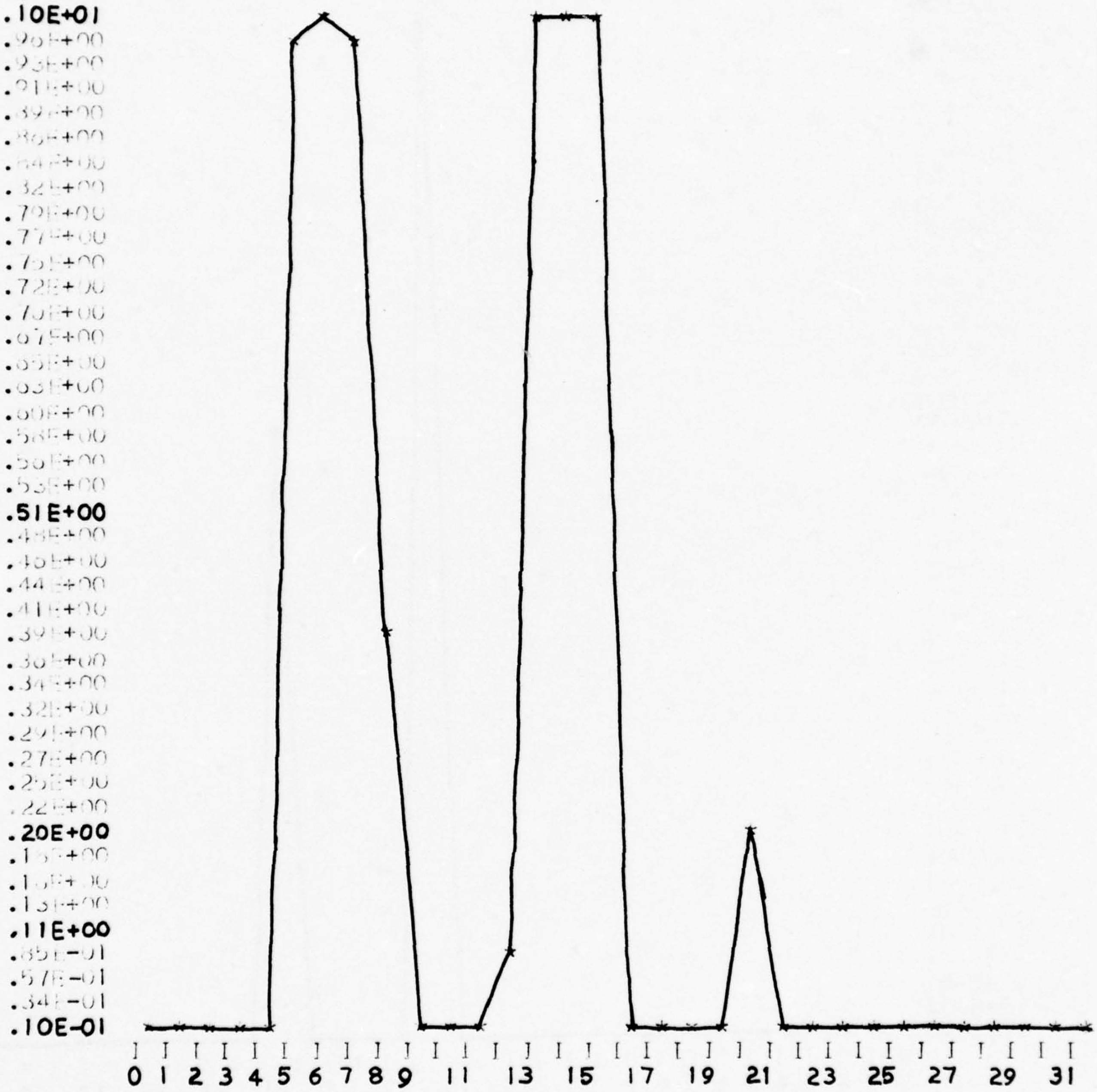
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 3
AFTER THE INFLUENCE OF CHANNEL 4 HAS BEEN REMOVED



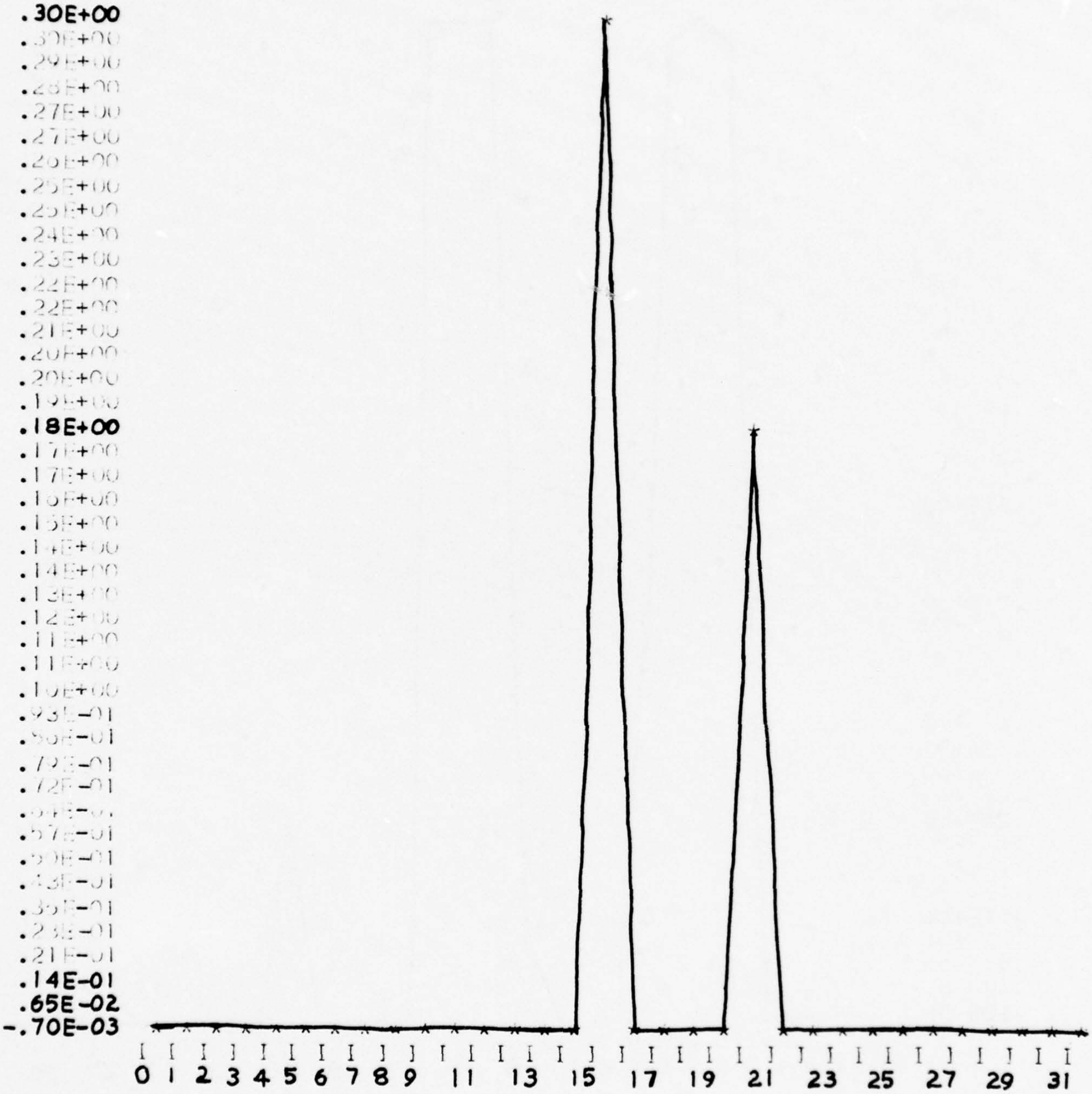
COHERENCE FOR CHANNELS 2 AND 4



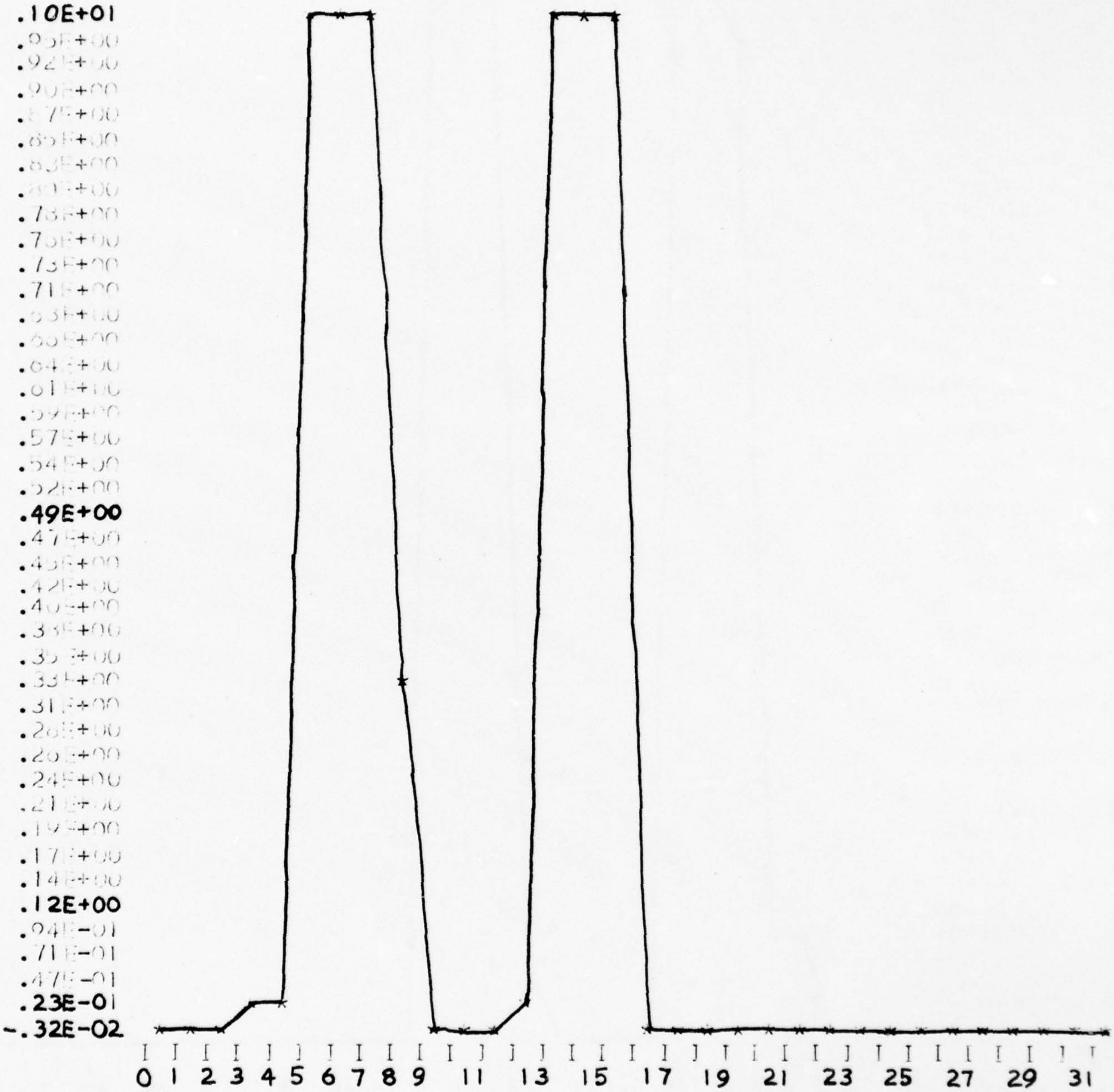
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 4
AFTER THE INFLUENCE OF CHANNEL 1 HAS BEEN REMOVED



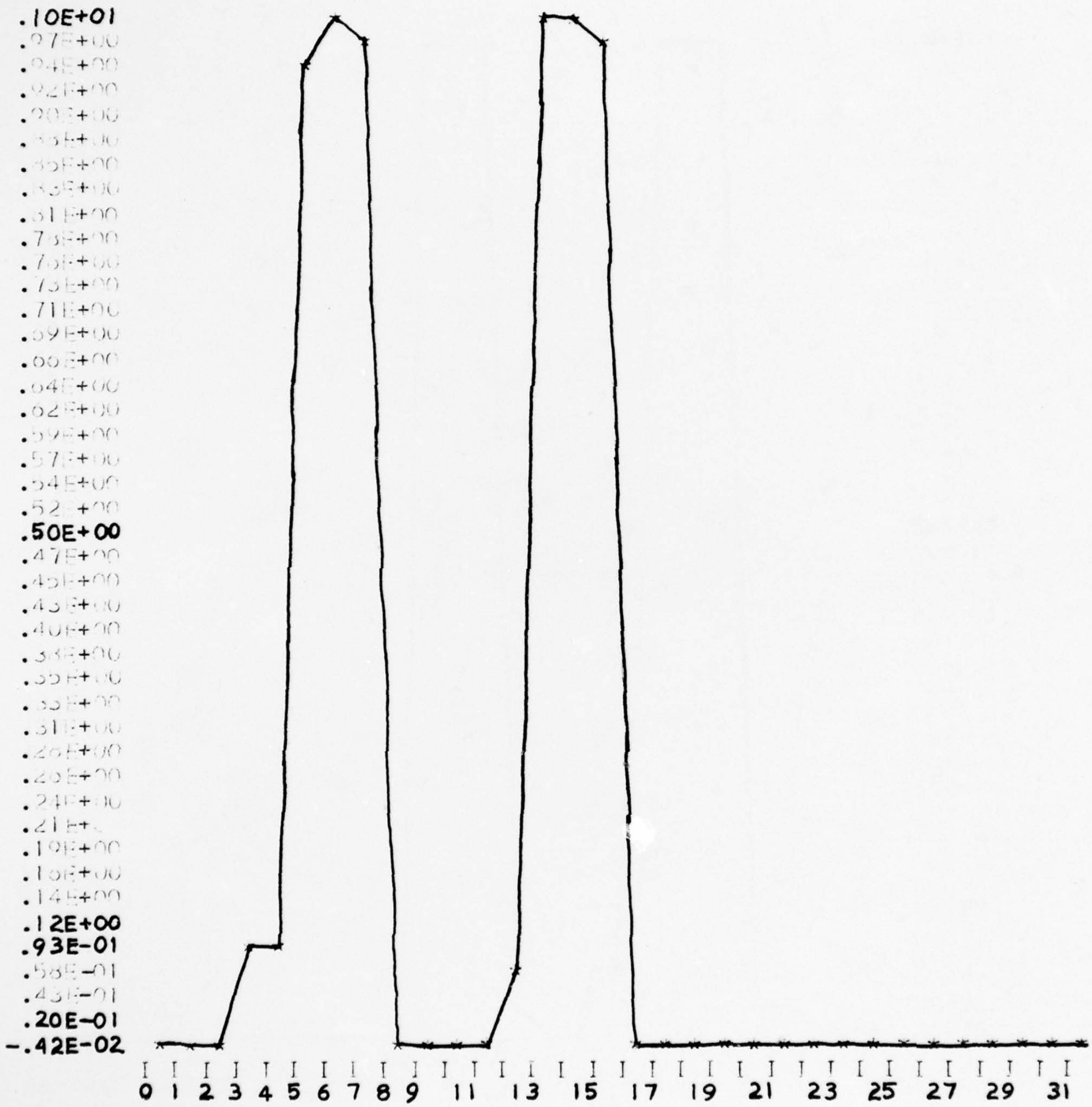
PARTIAL COHERENCE BETWEEN CHANNELS 2 AND 4
AFTER THE INFLUENCE OF CHANNEL 3 HAS BEEN REMOVED



COHERENCE FOR CHANNELS 3 AND 4

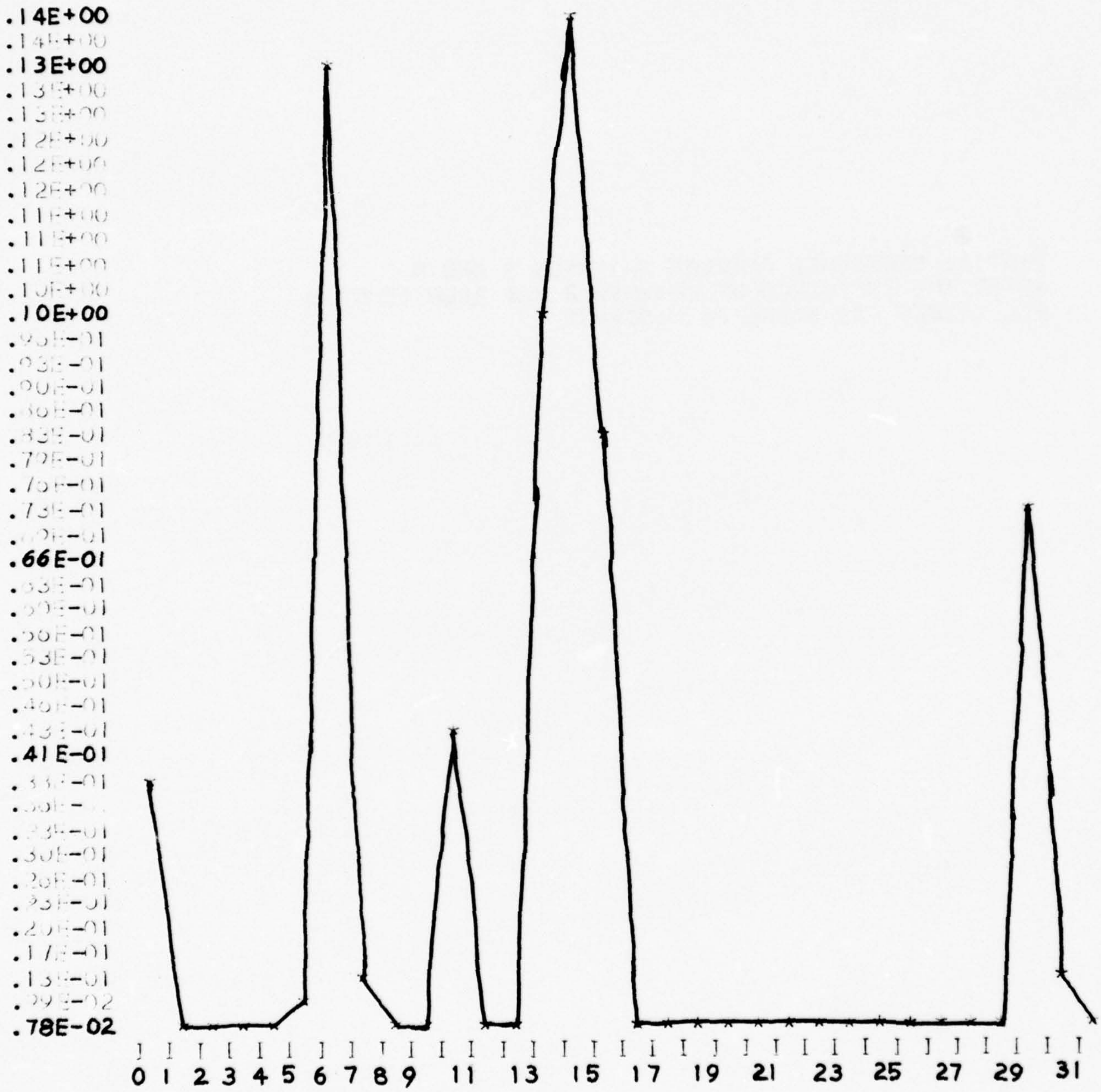


PARTIAL COHERENCE BETWEEN CHANNELS 3 AND 4
AFTER THE INFLUENCE OF CHANNEL 1 HAS BEEN REMOVED

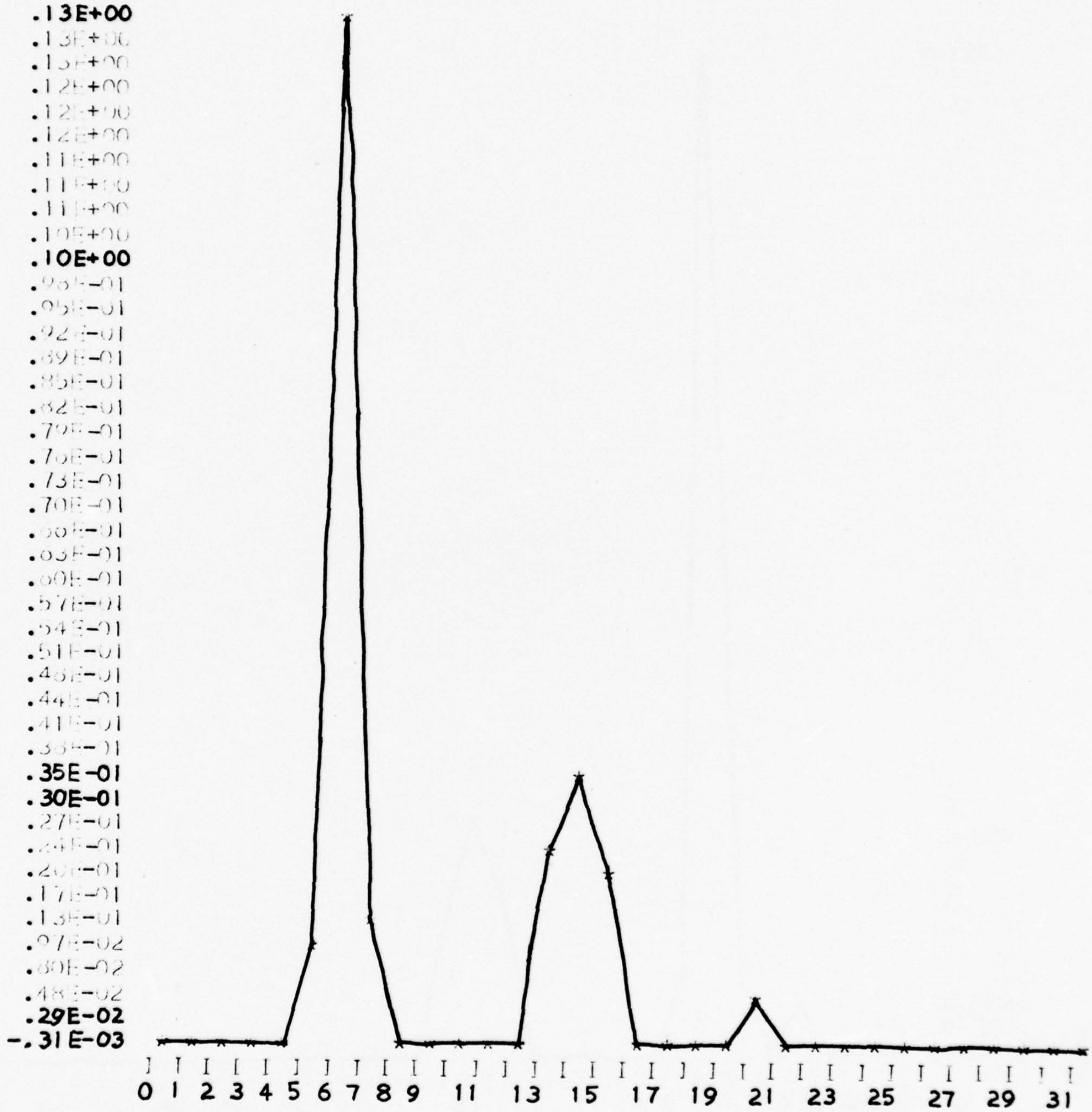


PARTIAL COHERENCE BETWEEN CHANNELS 3 AND 4
AFTER THE INFLUENCE OF CHANNEL 2 HAS BEEN REMOVED
ALL VALUES ARE EQUAL TO .00E-21

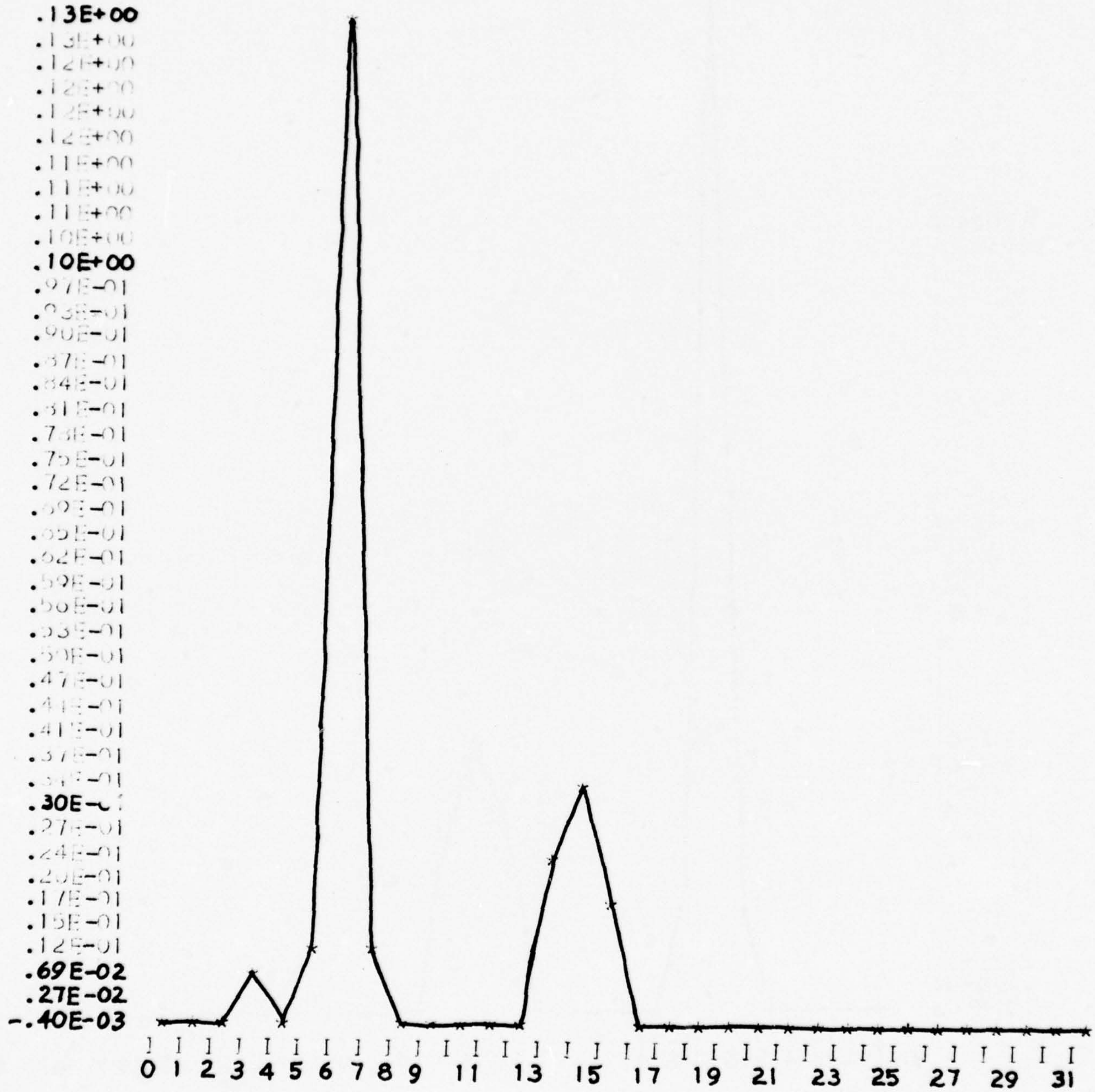
AUTOSPECTRA FOR CHANNEL 1



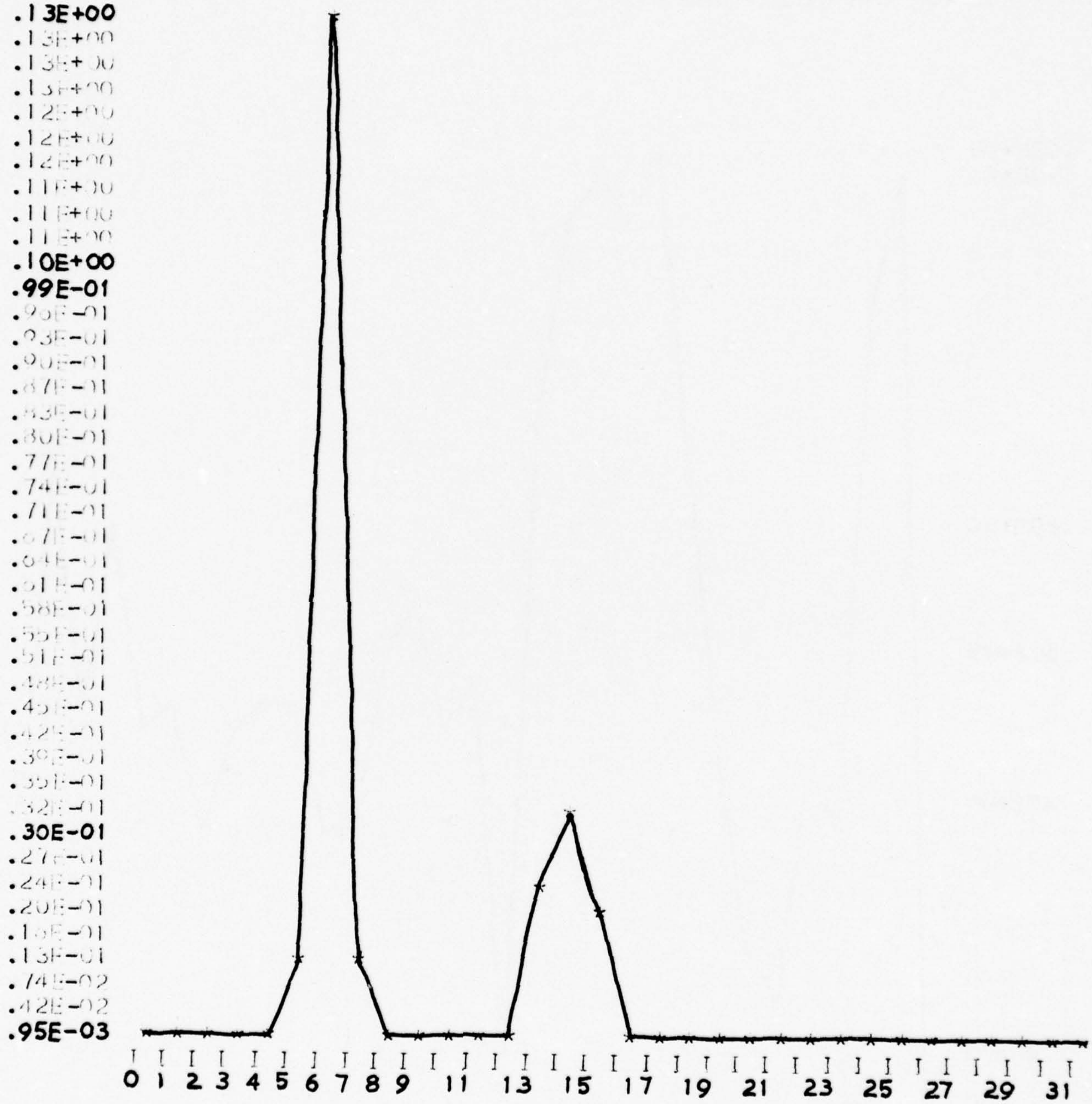
AUTOSPECTRA FOR CHANNEL 2



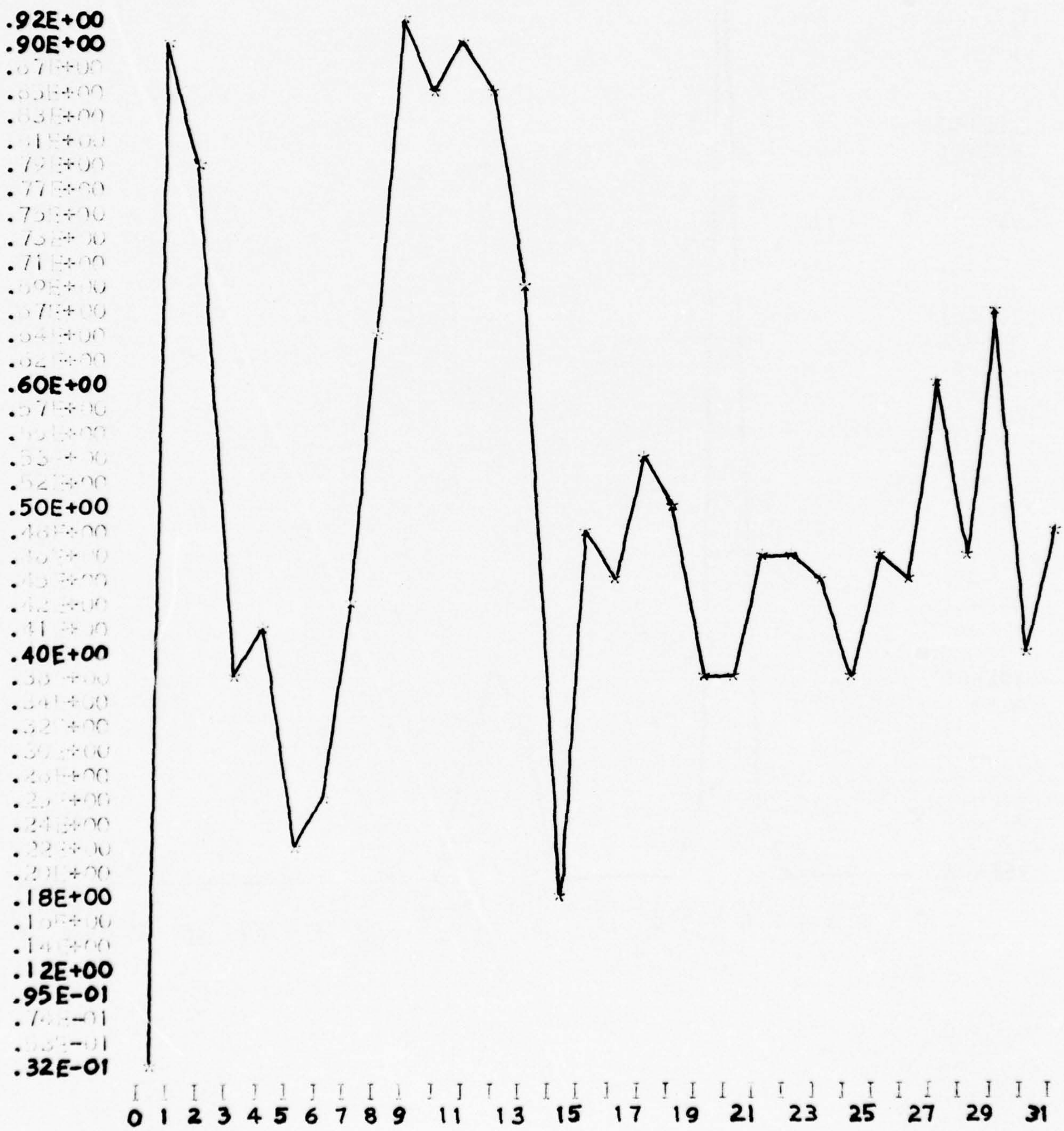
AUTOSPECTRA FOR CHANNEL 3



AUTOSPECTRA FOR CHANNEL 4



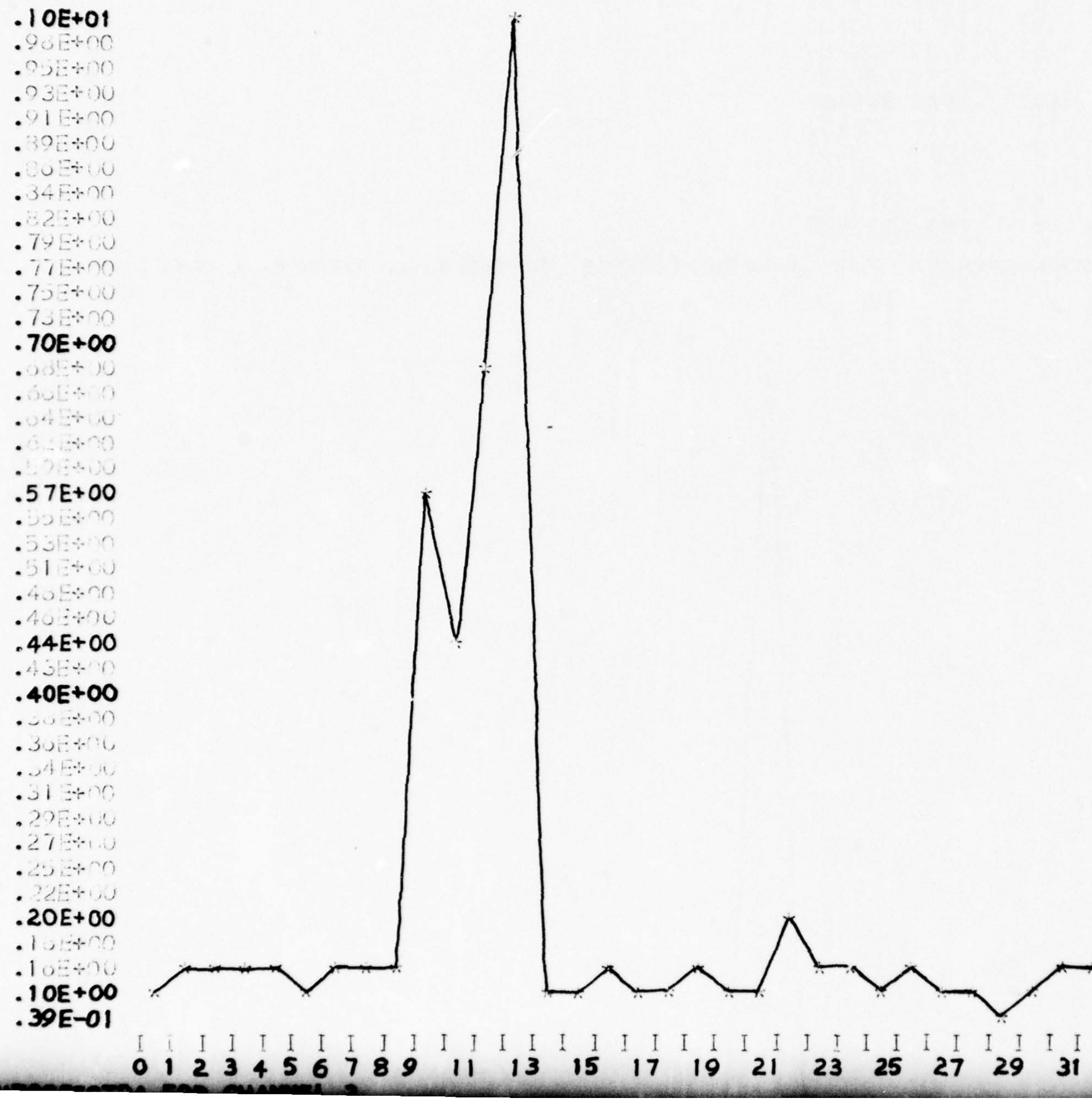
COHERENCE FOR CHANNELS 1 AND 2

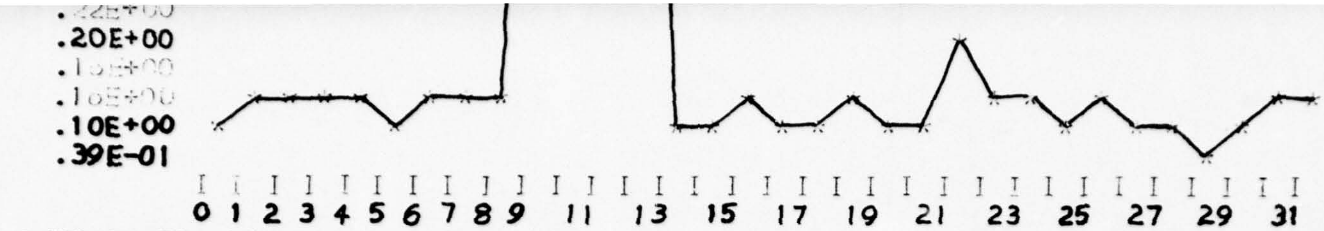


EEG COHERENCE ESTIMATE OBTAINED BY USING PROGRAM SPCTCLTK

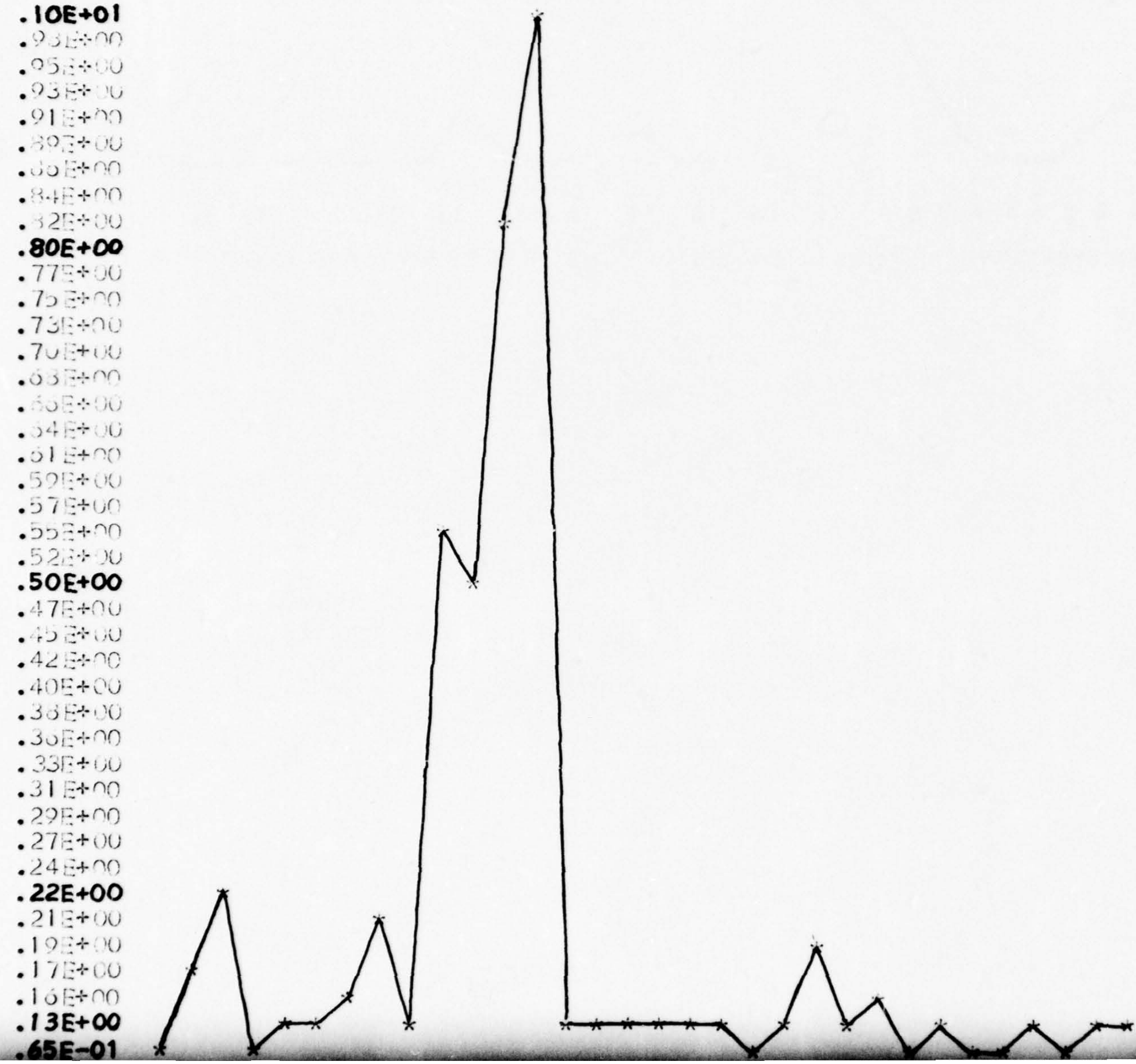
10 DEGREES OF FREEDOM

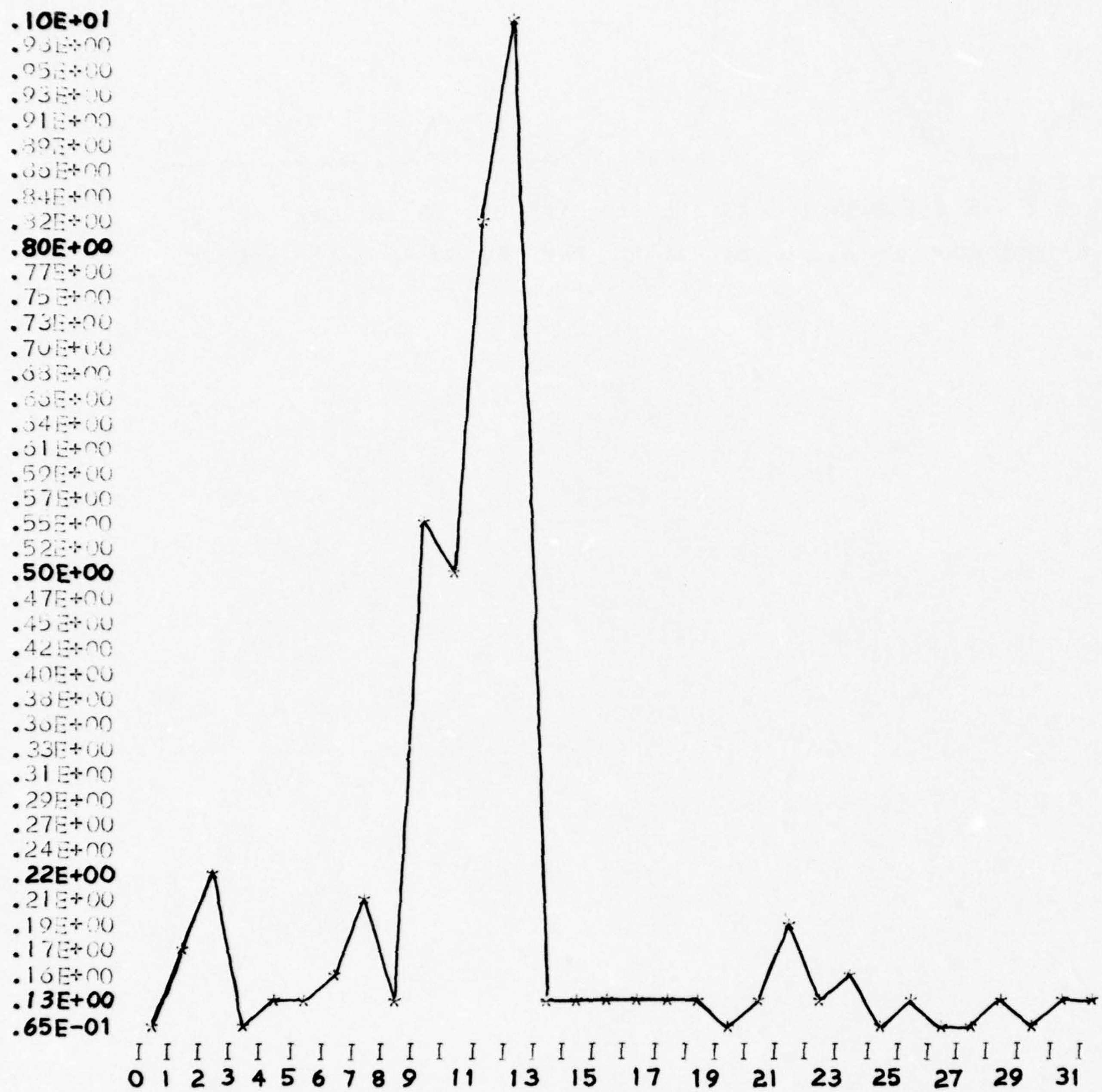
AUTOSPECTRA FOR CHANNEL 1





AUTOSPECTRA FOR CHANNEL 2

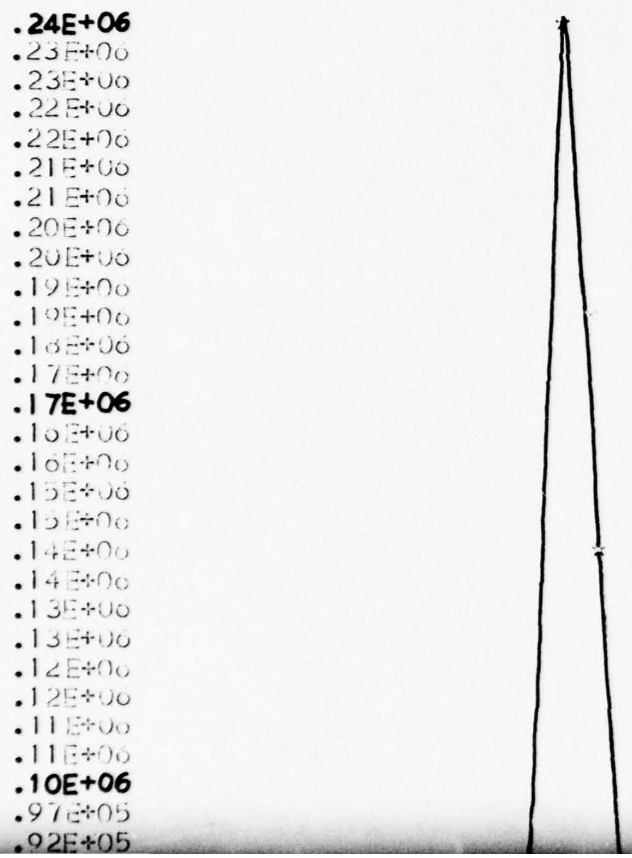


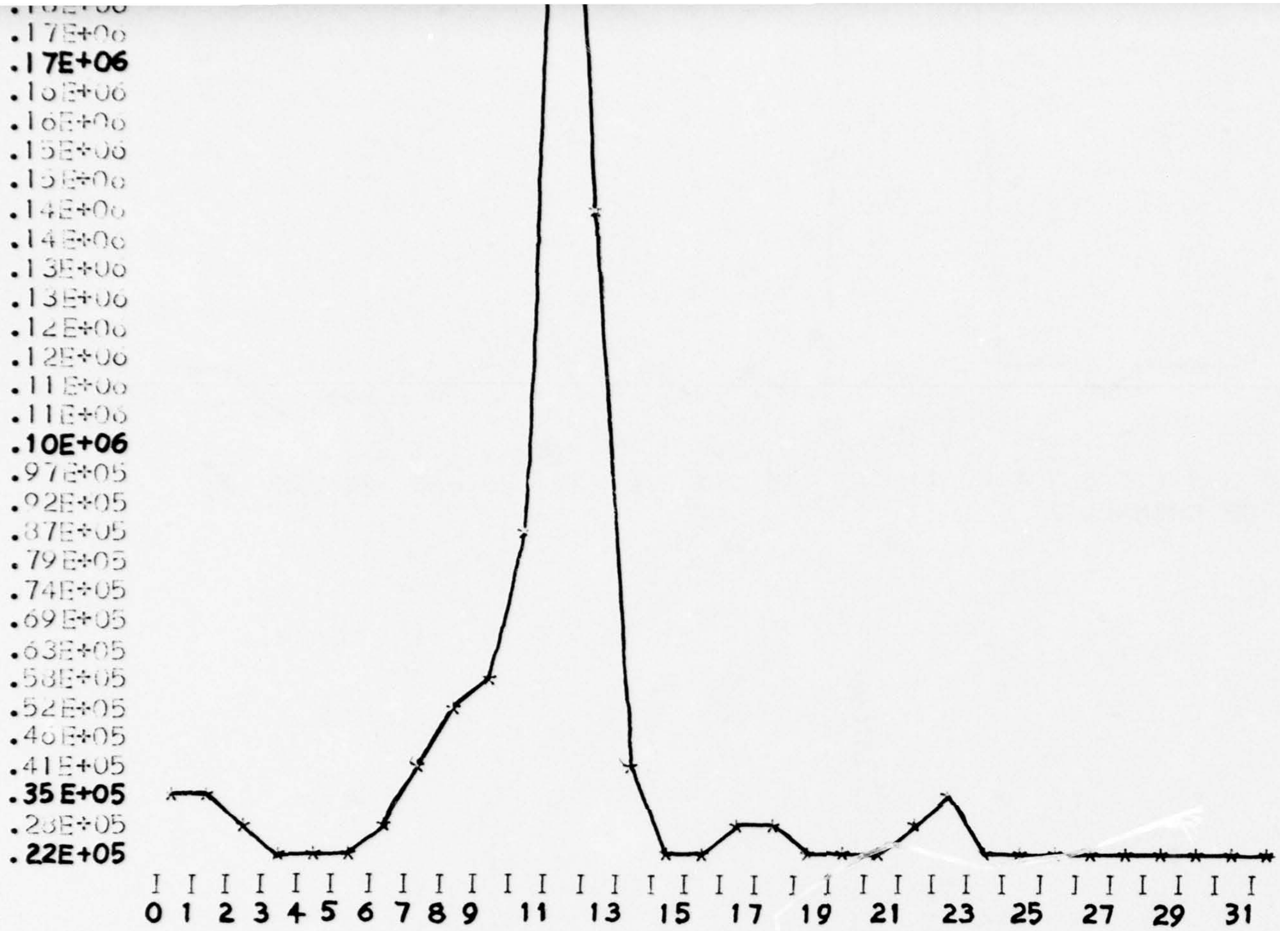


* EEG AUTO SPECTRA OBTAINED BY USING THE PROGRAM SPCTCLTK.
10 DEGREES OF FREEDOM

1	.2359876E+07
2	.2179372E+07
3	.2038137E+07
4	.2033934E+07
5	.1992783E+07
6	.1935647E+07
7	.1931619E+07
8	.1929990E+07
9	.1929988E+07
10	.1928289E+07
11	.1911185E+07
12	.1910493E+07
13	.1893981E+07
14	.1892531E+07
15	.1889173E+07

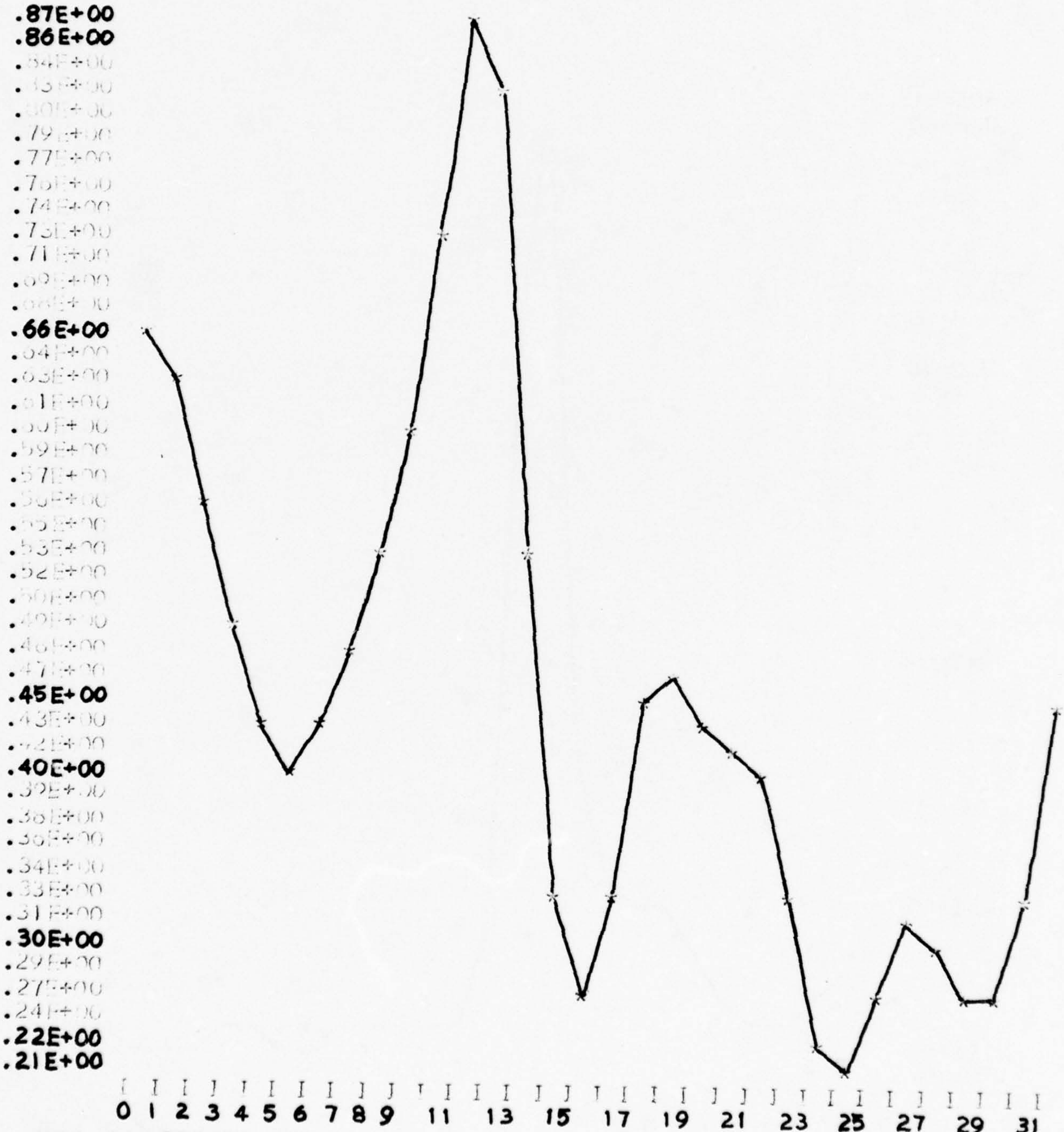
PREDICTION ERRORS FOR AUTOREGRESSIVE SCHEMES OF ORDER 1 TO 15





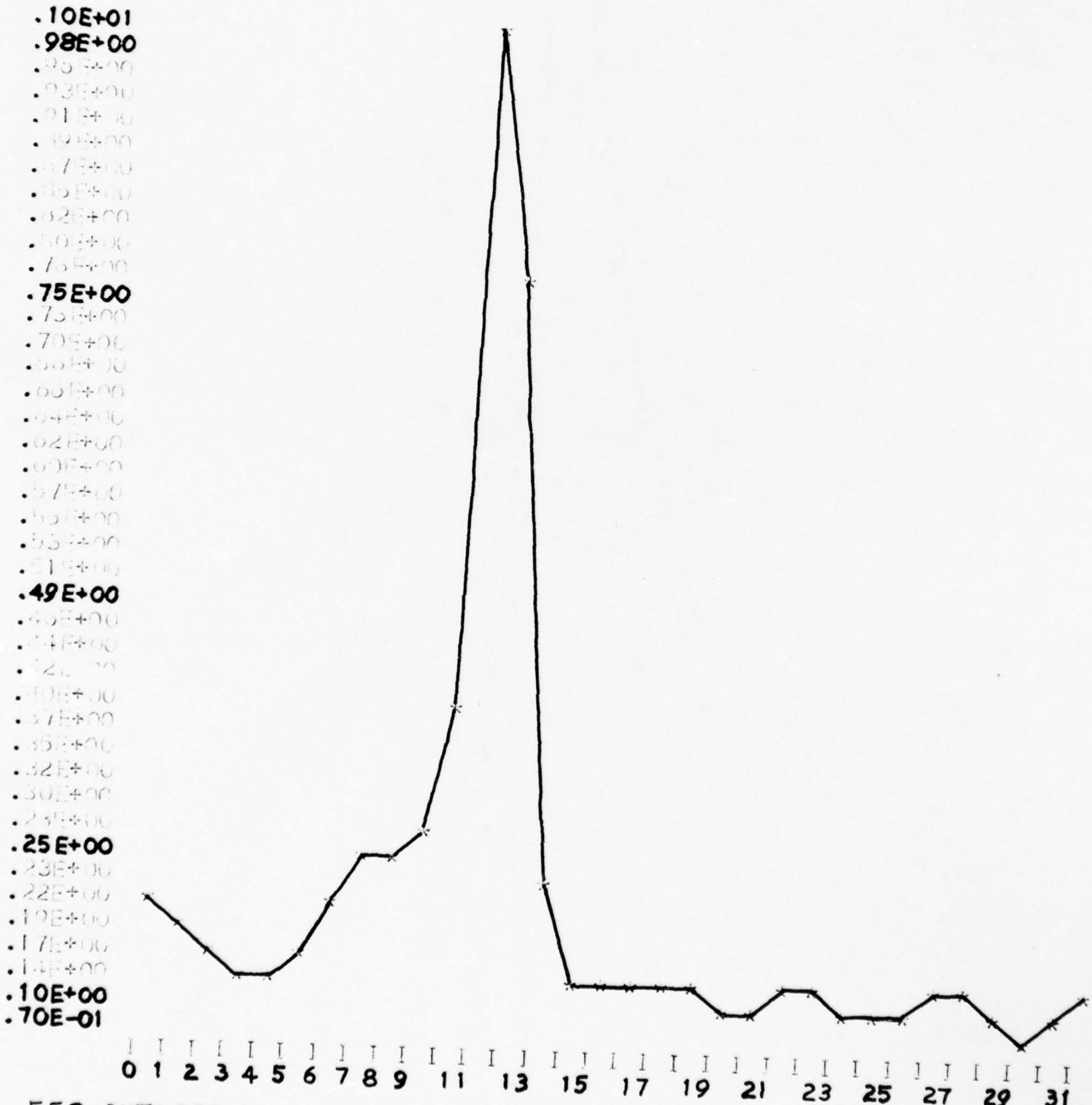
* EEG AUTO SPECTRUM OBTAINED BY USING THE PROGRAM AUTOREG

COHERENCE FOR CHANNELS 1 AND 2



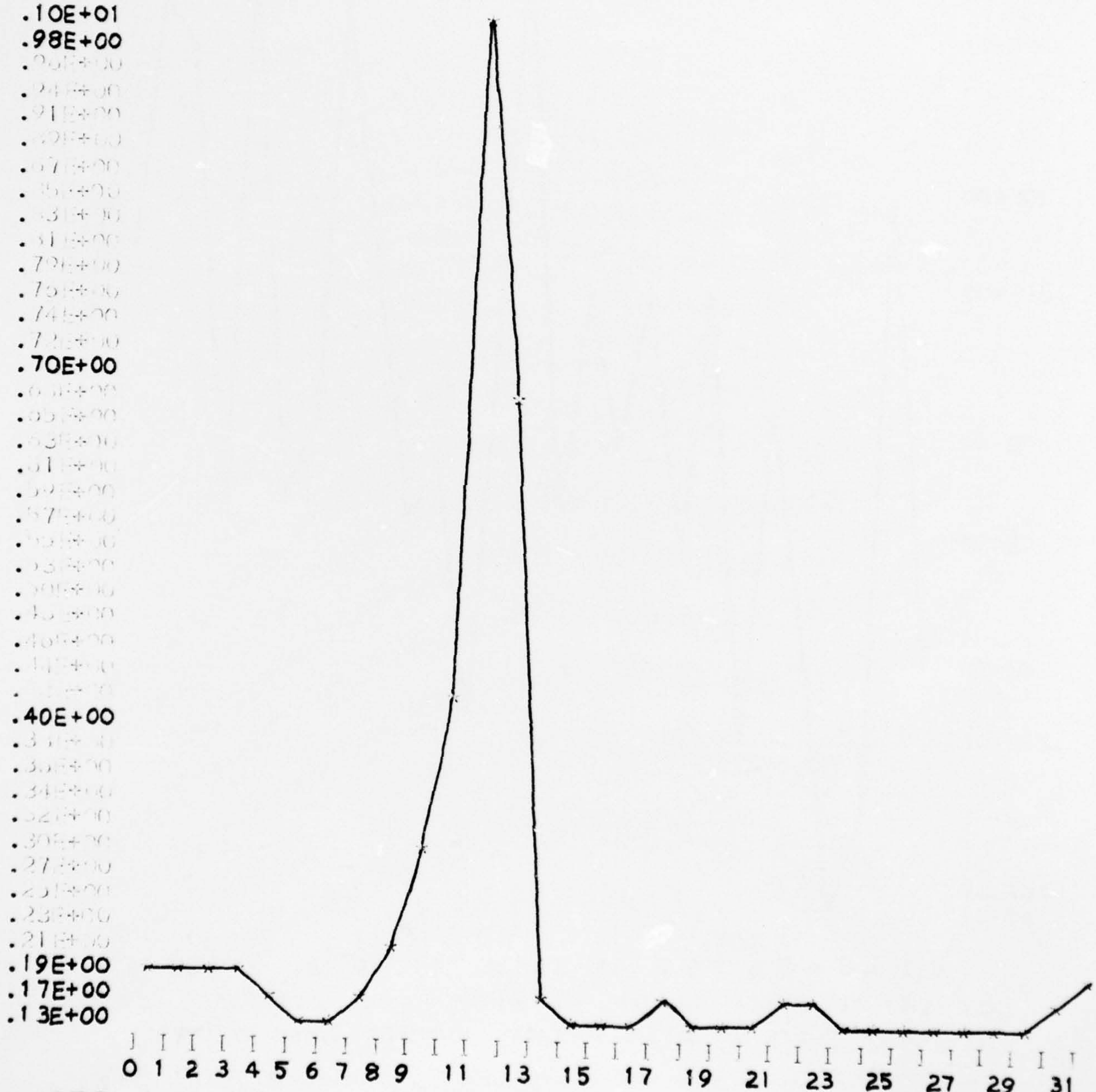
EEG COHERENCE COMPUTED BY USING PROGRAM MAUTOREG

AUTOSPECTRA FOR CHANNEL 1



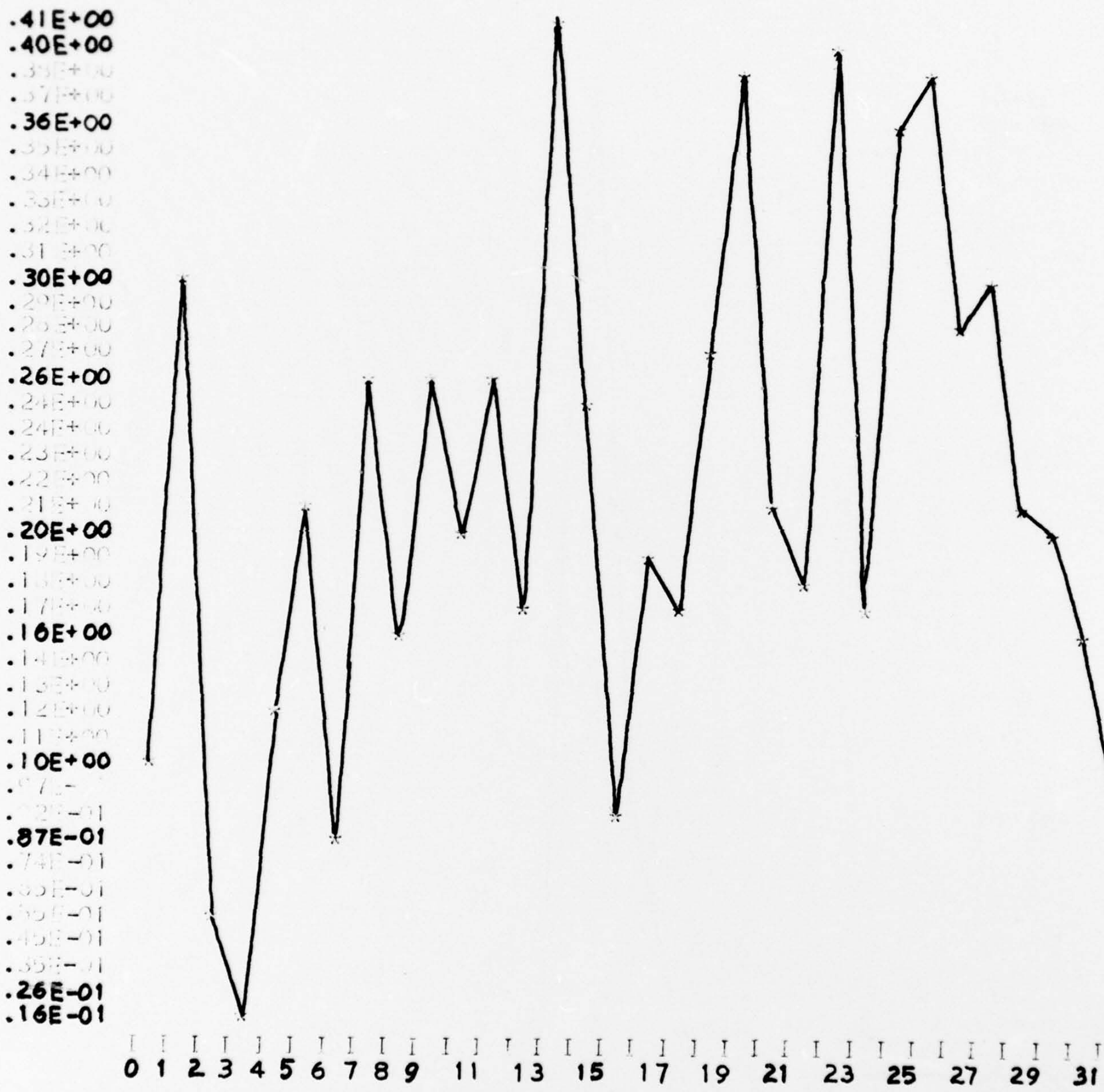
EEG AUTOSPECTRUM ESTIMATE OBTAINED BY USING THE PROGRAM MAUTOREG

AUTOSPECTRA FOR CHANNEL 2



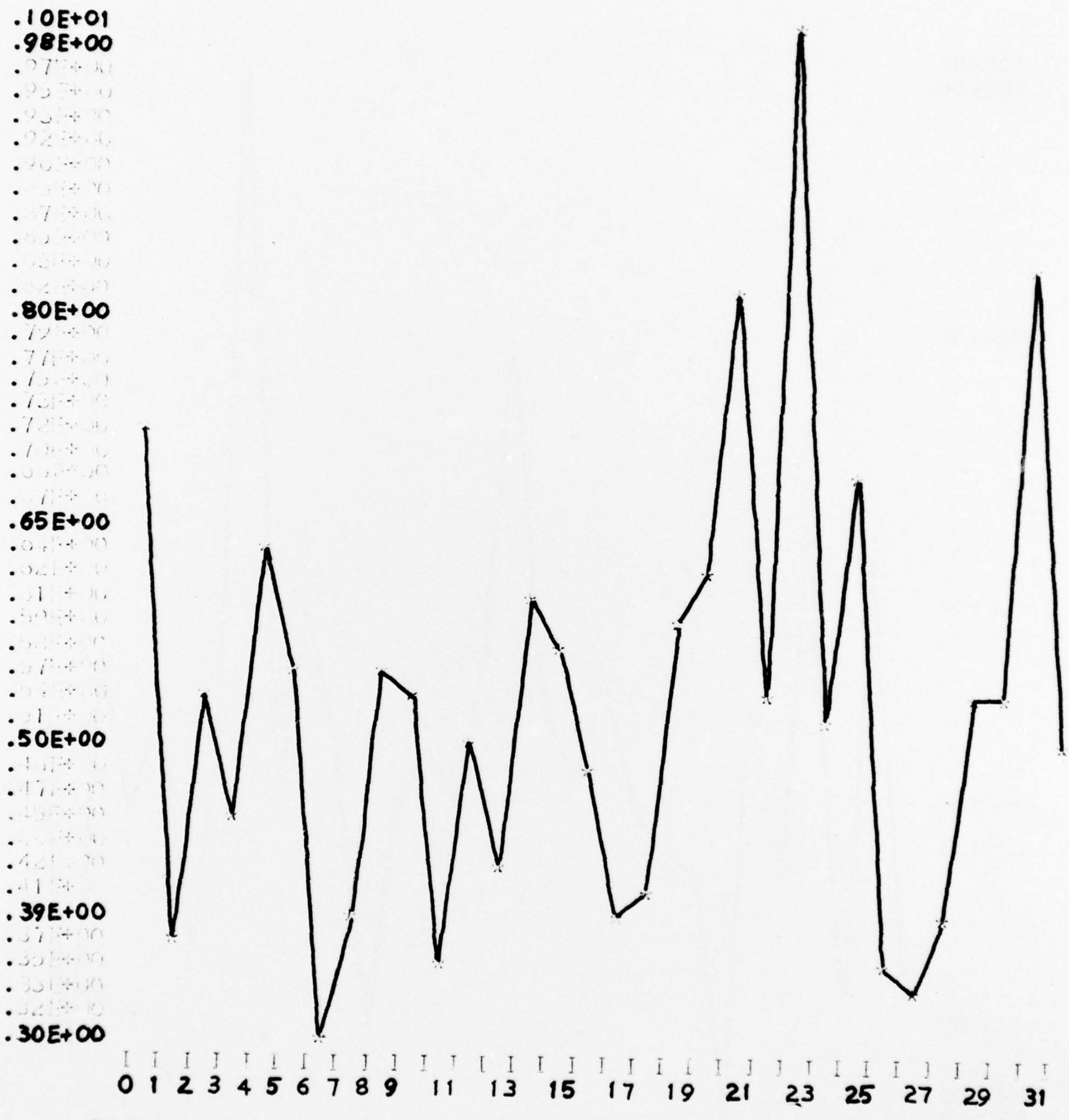
EEG AUTO SPECTRUM ESTIMATE OBTAINED BY USING THE PROGRAM MAUTOREG

COHERENCE FOR CHANNELS 1 AND 2



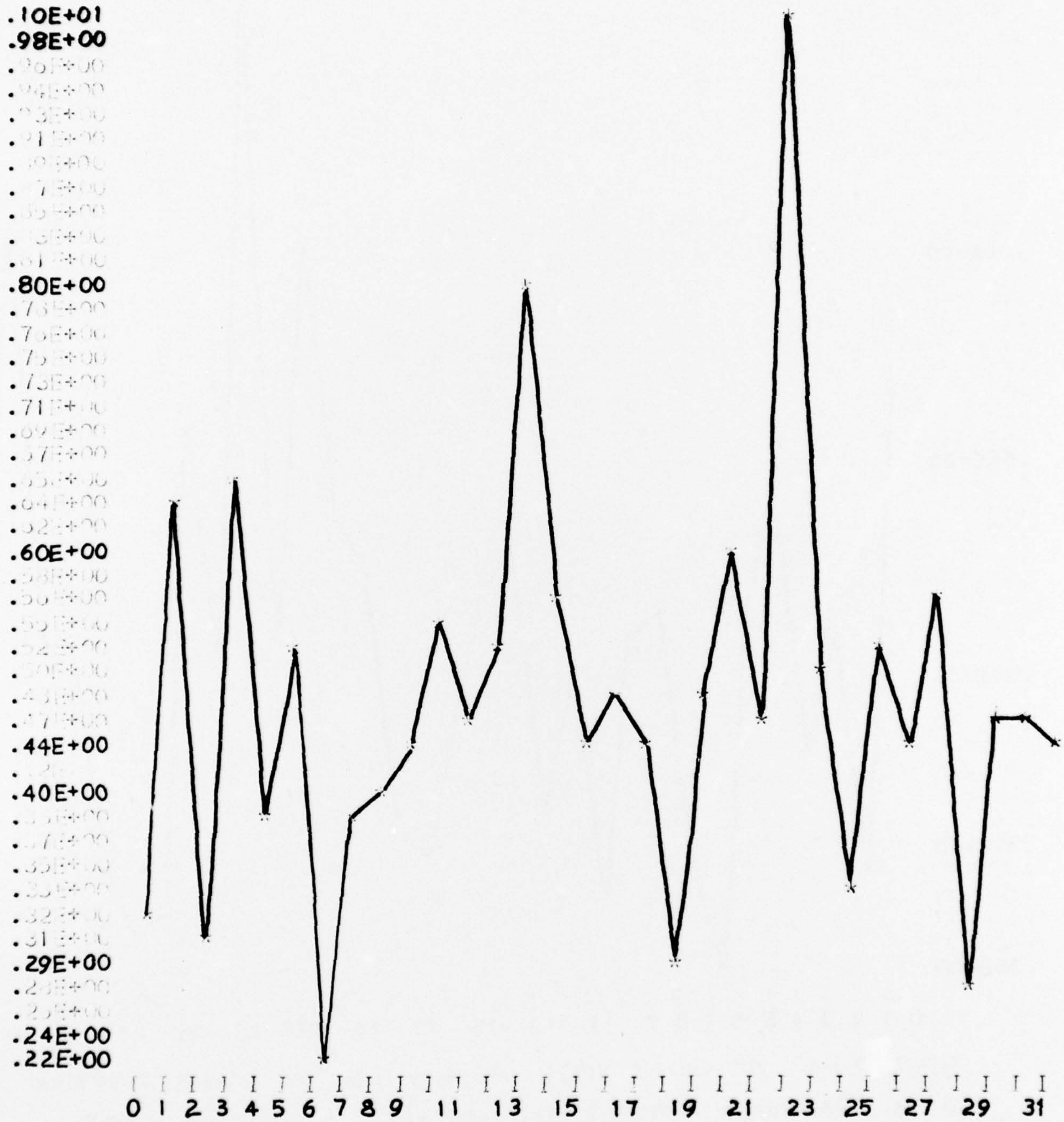
COHERENCE SPECTRUM OF TWO WHITE NOISE PROCESSES OBTAINED BY USING FORTRAN DTSS RANDOM NUMBER GENERATOR.
CHANNELS ARE HIGHLY COHERENT, SHOWING POOR PERFORMANCE OF RANDOM NUMBER GENERATOR.
ESTIMATES WERE COMPUTED BY USING PROGRAM WNOISPEC.
31 DEGREES OF FREEDOM.

AUTOSPECTRA FOR CHANNEL 1



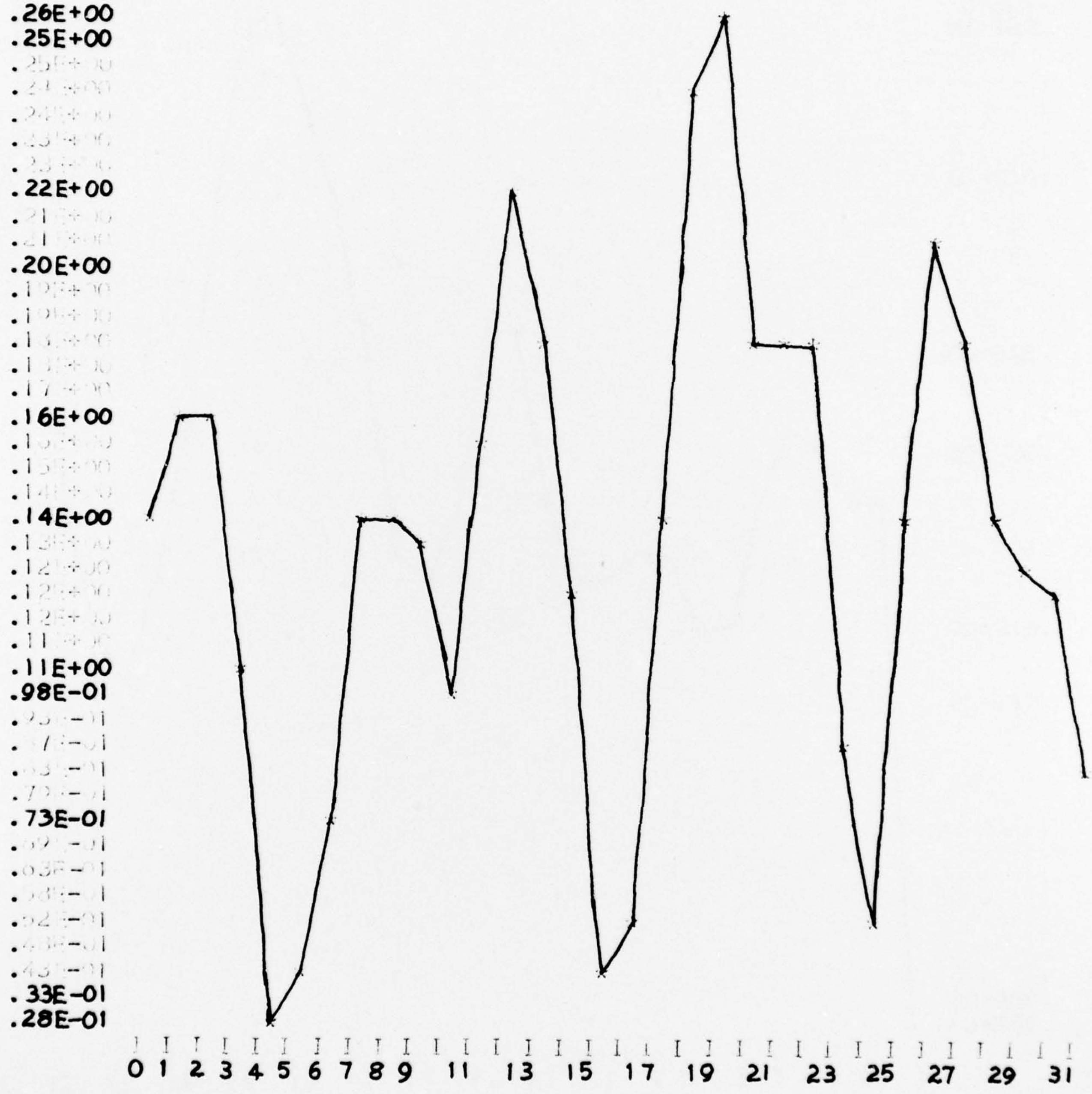
SPECTRUM OF WHITE NOISE GENERATED BY USING FORTRAN DTSS RANDOM NUMBER GENERATOR. ESTIMATES WERE OBTAINED BY USING PROGRAM WNOISPEC. 31 DEGREES OF FREEDOM WERE USED. STATISTICAL ANALYSIS OF SPECTRUM SHOWS POOR PERFORMANCE OF RANDOM NUMBER GENERATOR.

AUTOSPECTRA FOR CHANNEL 2



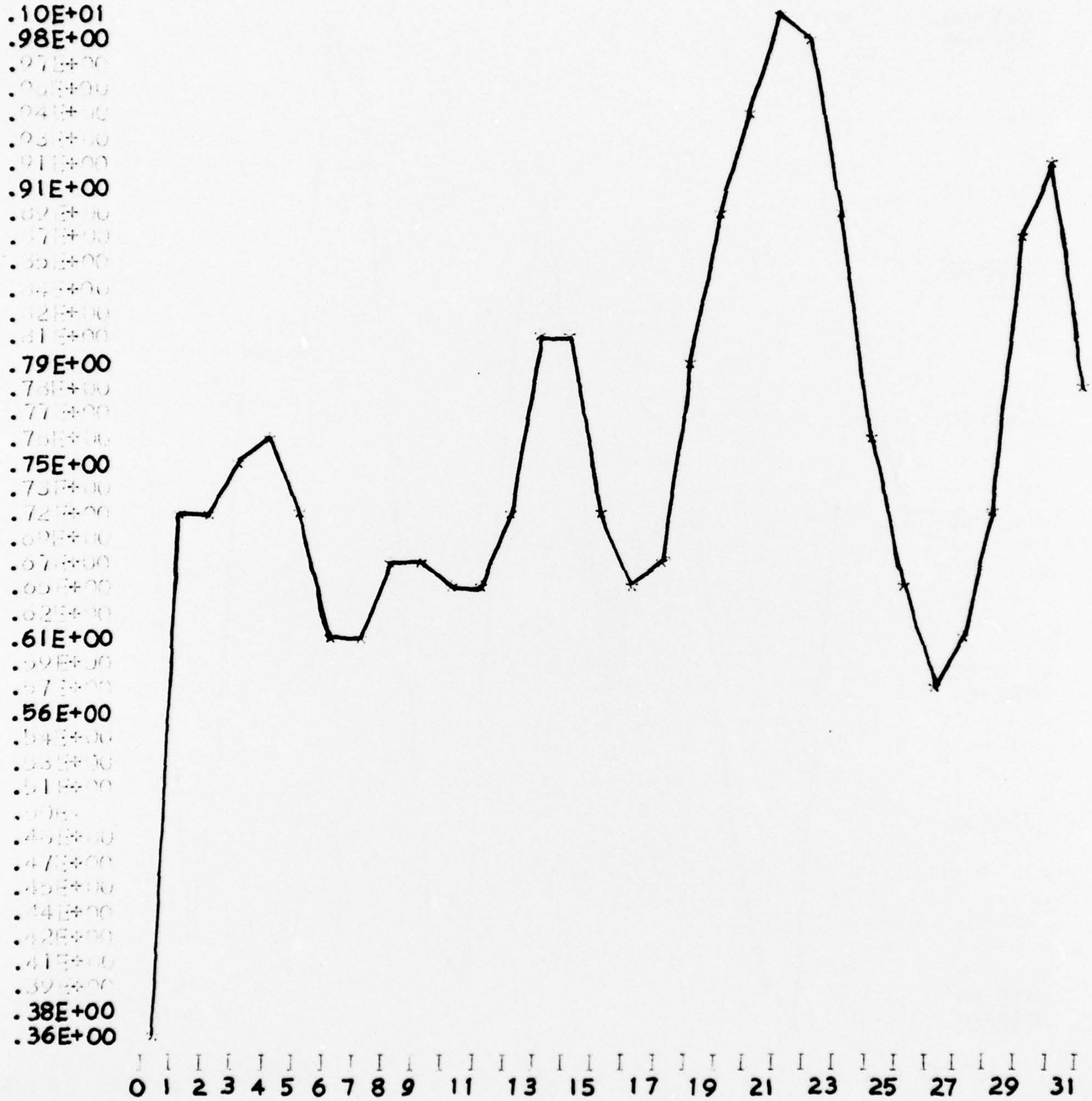
SPECTRUM FOR WHITE NOISE PROCESS OBTAINED BY USING A RANDOM NUMBER GENERATOR. PROGRAM WNOISPEC WAS USED, WITH 31 DEGREES OF FREEDOM. TEST SHOWS POOR PERFORMANCE OF GENERATOR.

COHERENCE FOR CHANNELS 1 AND 2



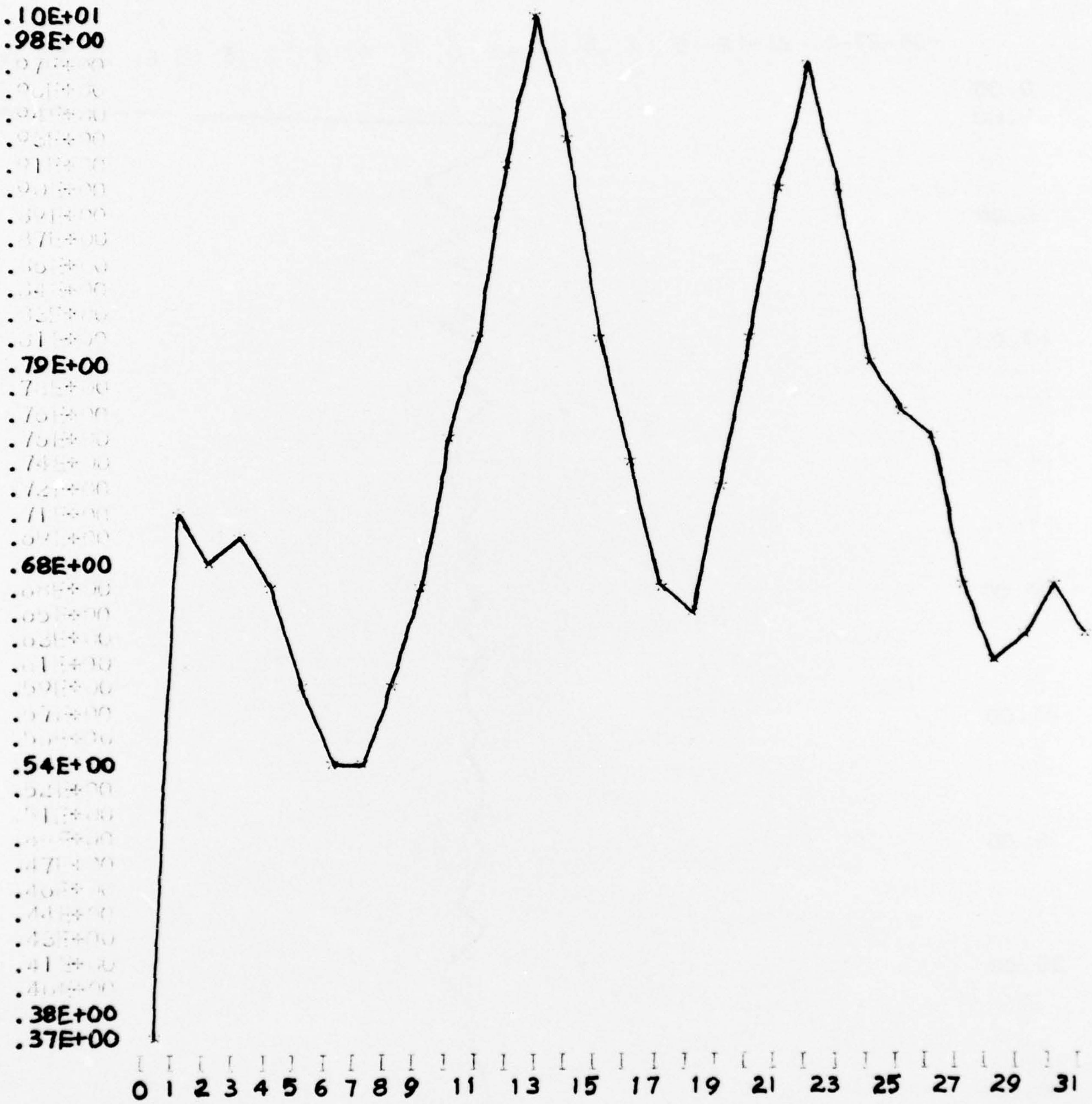
COHERENCE FUNCTION FOR TWO WHITE NOISE PROCESSES OBTAINED BY USING DTSS FORTRAN RANDOM NUMBER GENERATOR. PROGRAM WNOITEST WAS USED WITH 116 DEGREES OF FREEDOM. TEST SHOWS POOR PERFORMANCE OF RANDOM NUMBER GENERATOR.

AUTOSPECTRA FOR CHANNEL 1



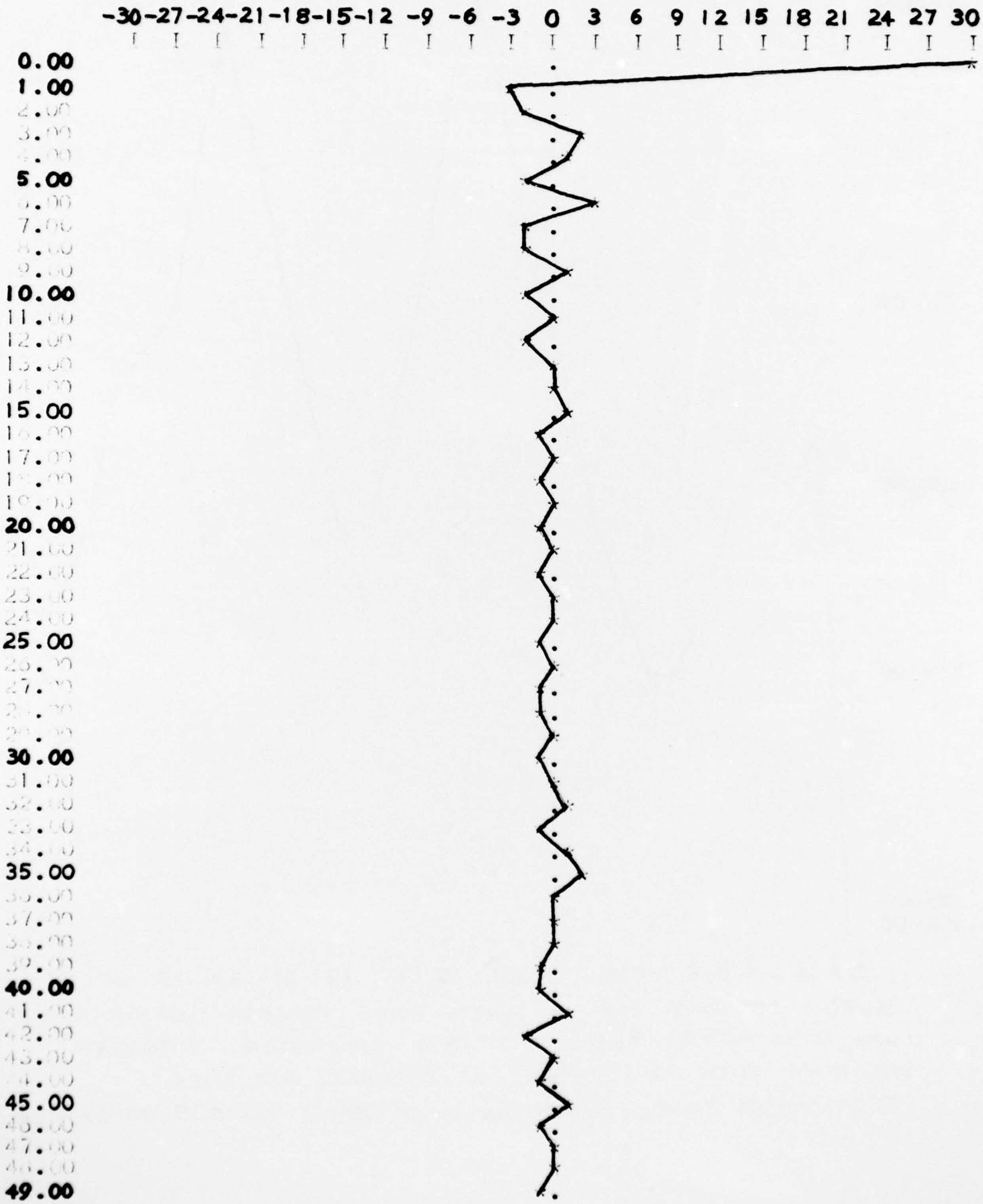
AUTO SPECTRUM OF WHITE NOISE PROCESS OBTAINED BY USING DTSS FORTRAN RANDOM NUMBER GENERATOR. PROGRAM WNOITEST, WITH 116 DEGREES OF FREEDOM WAS USED. TEST SHOWS POOR PERFORMANCE OF RAND. NUMBER GENERATOR.

AUTOSPECTRA FOR CHANNEL 2



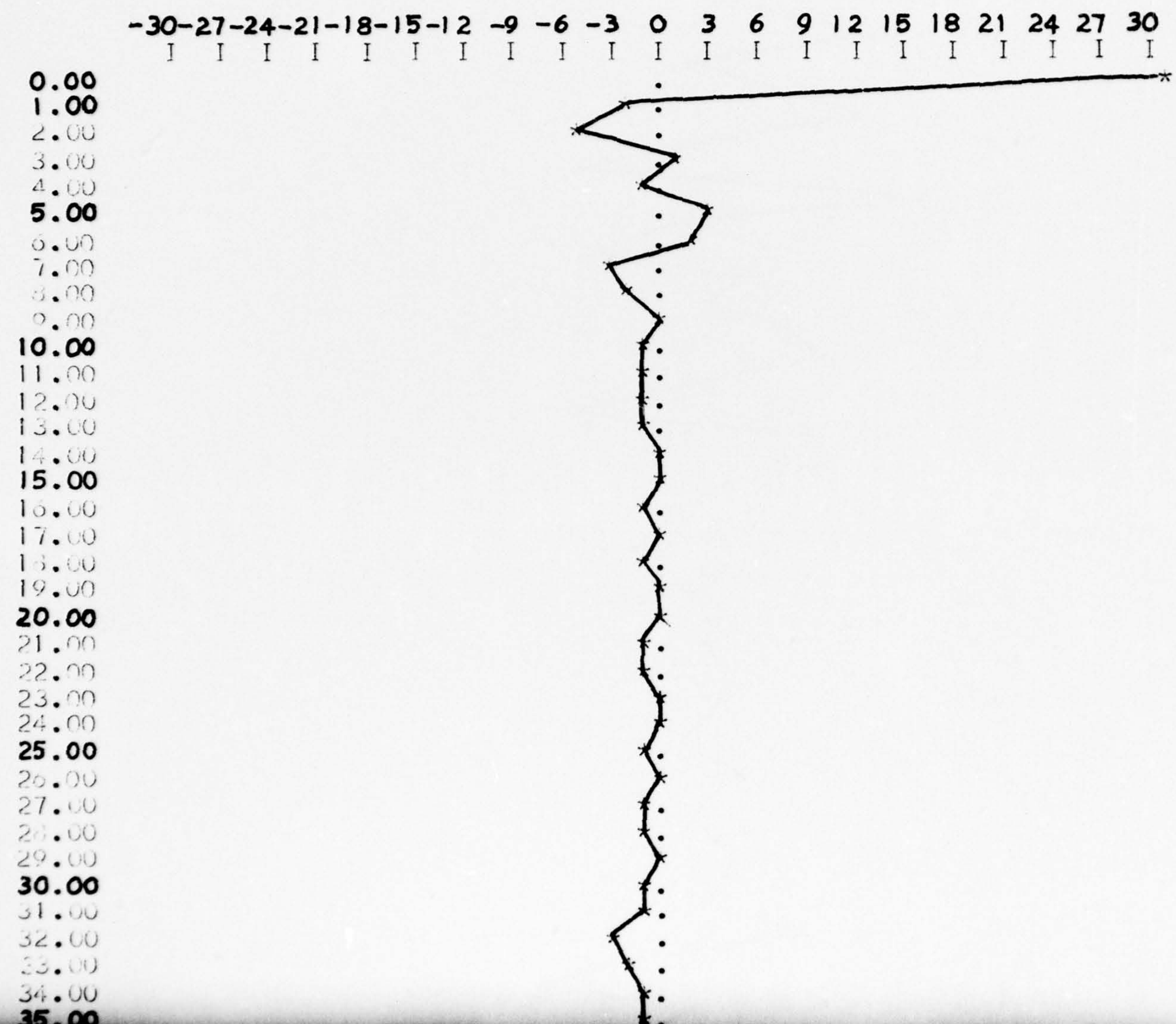
AUTO SPECTRUM FOR A WHITE NOISE PROCESS OBTAINED BY USING DTSS FORTRAN RANDOM NUMBER GENERATOR. PROGRAM WNOITEST, WITH 116 DEGREES OF FREEDOM WAS USED.
TEST SHOWS POOR PERFORMANCE OF RAND. NUMBER GENERATOR.

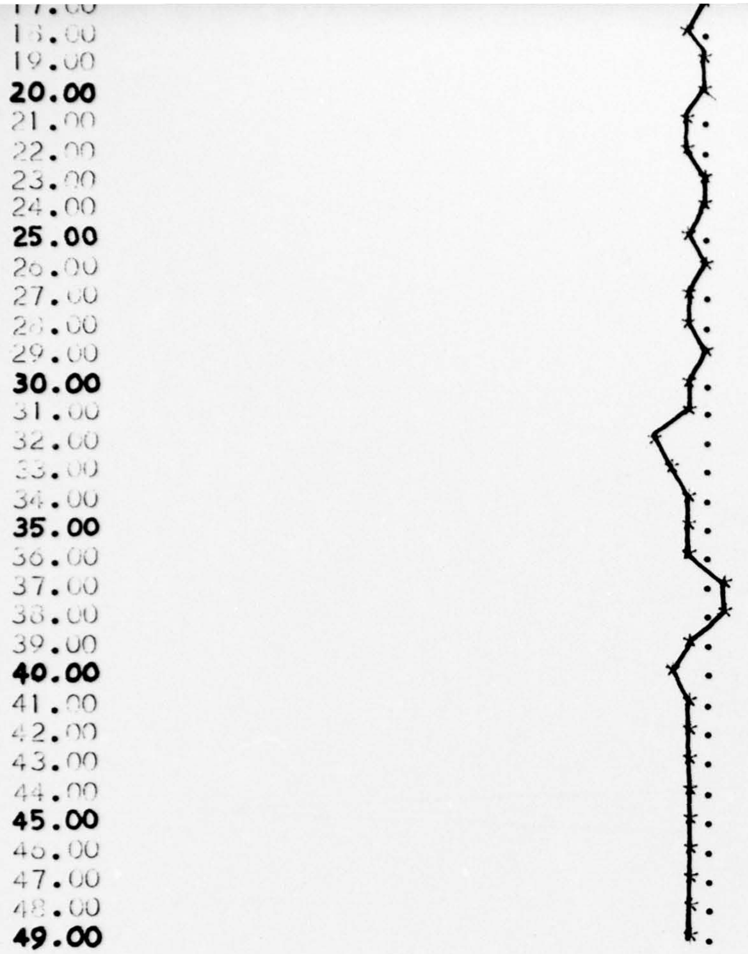
AUTO CORRELATION FOR CHANNEL 1
LARGEST VALUE IS .6915590E+02
SMALLEST VALUE IS -.5173850E+01
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .2230835E+01



AUTOCORRELATION FOR WHITE NOISE PROCESS

AUTO CORRELATION FOR CHANNEL 2
LARGEST VALUE IS .6397735E+02
SMALLEST VALUE IS -.8328822E+01
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .2063785E+01

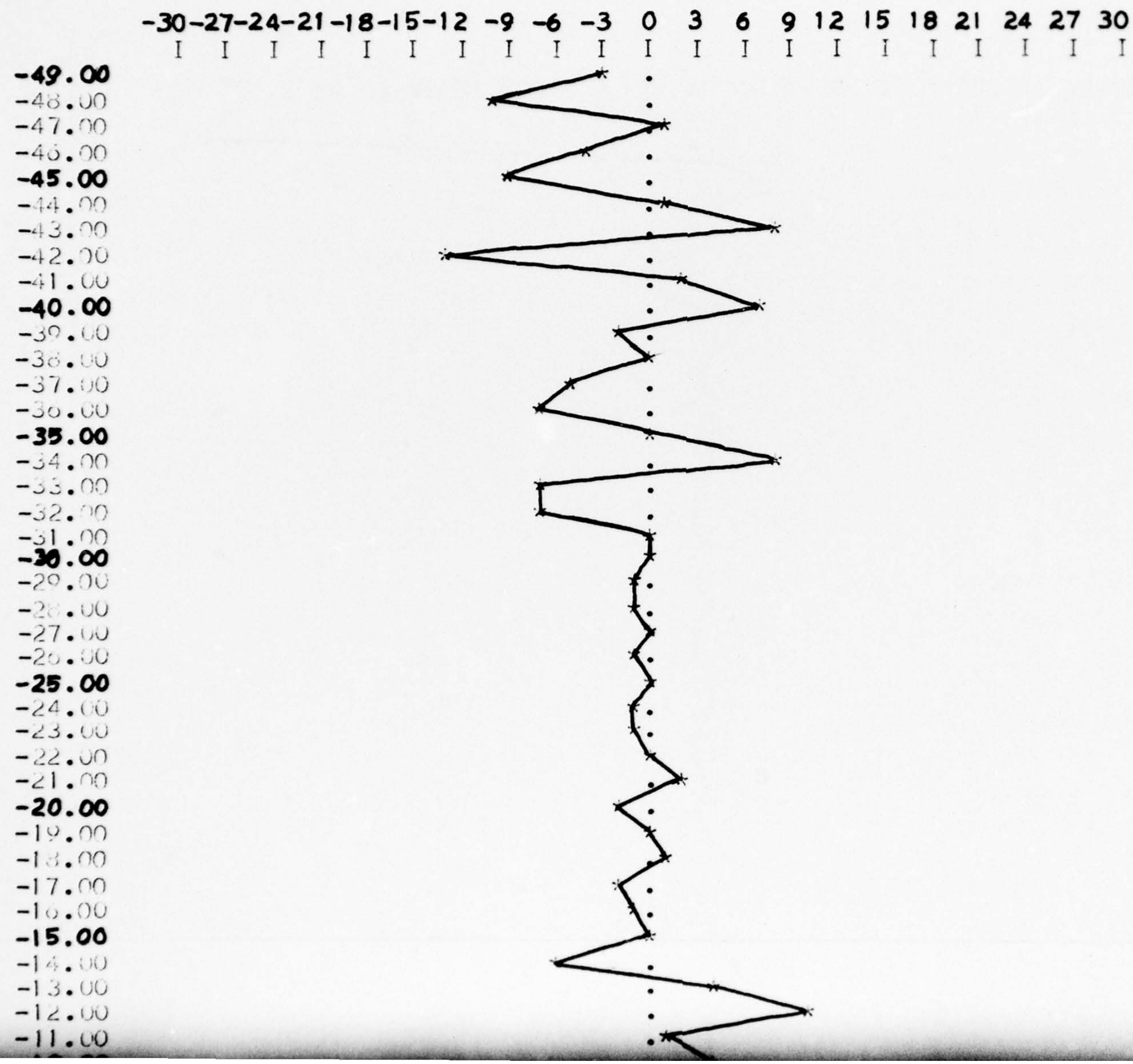




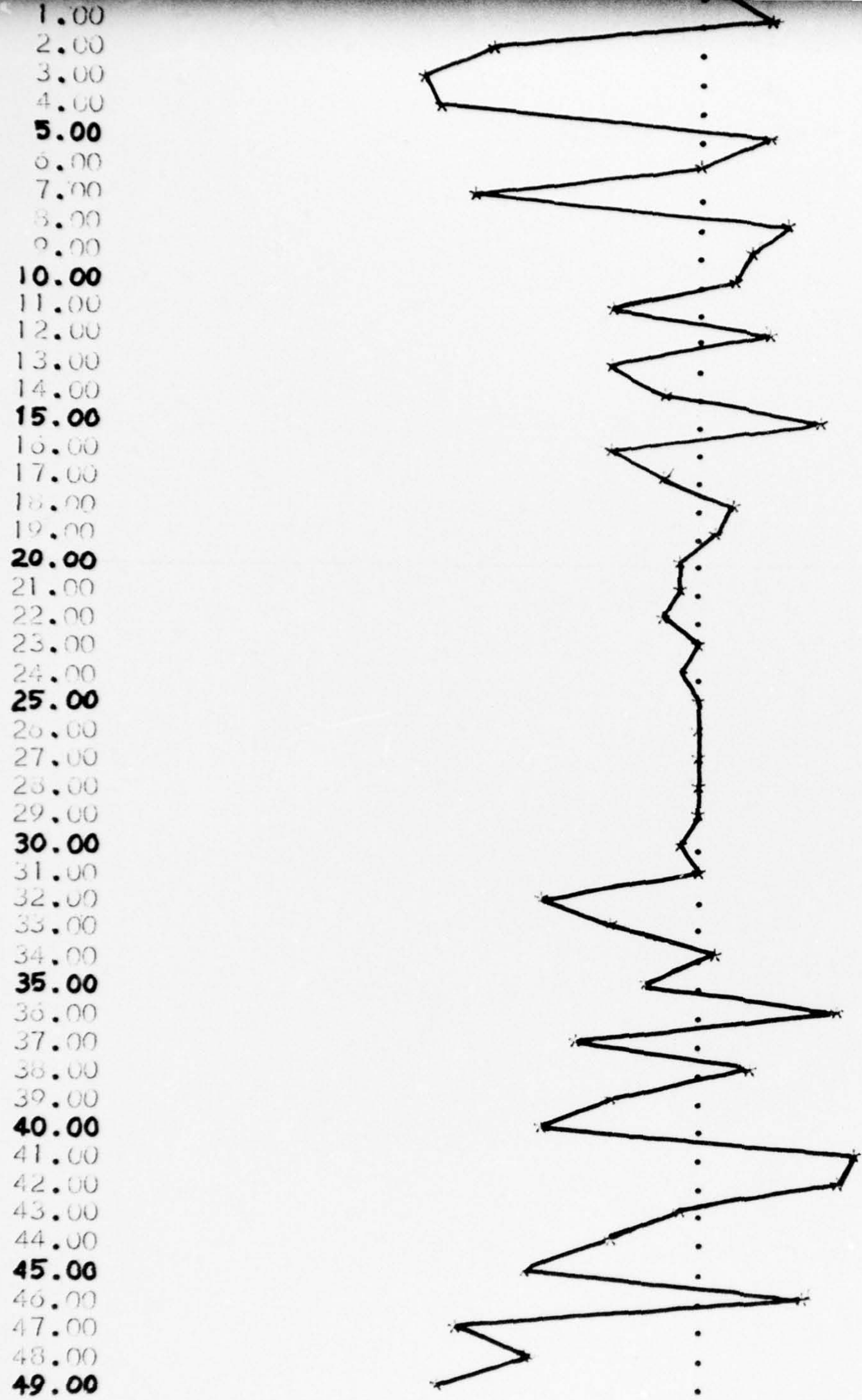
AUTO CORRELATION FOR WHITE NOISE PROCESS.

CROSS CORRELATION BETWEEN CHANNELS 1 AND 2
LARGEST VALUE IS .8103186E+01
SMALLEST VALUE IS -.4139597E+01
MULTIPLY EACH ORDINATE READING BY THE SCALE FACTOR .2613931E+00

152




-17.00
-16.00
-15.00
-14.00
-13.00
-12.00
-11.00
-10.00
-9.00
-8.00
-7.00
-6.00
-5.00
-4.00
-3.00
-2.00
-1.00
0.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00
17.00
18.00
19.00
20.00
21.00
22.00
23.00
24.00
25.00
26.00
27.00
28.00
29.00
30.00
31.00
32.00
33.00
34.00
35.00
36.00
37.00



CROSS CORRELATION FOR TWC WHITE NOISE PROCESSES OBTAINED BY USING RANDOM NUMBER GEKERATOR. GENERATOR PERFORMS POORLY.

3

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) U.S. Naval Academy, Annapolis.	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

Mathematical -statistical and digital computer analysis of time series data.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research report.

5. AUTHOR(S) (First name, middle initial, last name)

Steven J. Raher.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
17 May 1976.	152	15

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) U.S. Naval Academy, Annapolis - Trident Scholar project report, no. 81 (1976) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release; its distribution is UNLIMITED.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Naval Academy, Annapolis.

13. ABSTRACT

The purpose of this study is to develop a series of computer programs for use in analyzing time series data (waves) - such as EEG readings.

Programs were used to produce reliable estimates of correlations, spectra, cross spectra, and partial coherences of multi-channel random processes The software package was written to be easily adaptable to different sampling rates, amounts of data, and numbers of channels. Provisions for digital pre-filtering of data, detrending, and smoothing (using a number of lag windows) were also included. Techniques for estimation of spectra by fitting single- and multi-channel autoregressive schemes to sampled data were also applied and found to yield results consistent with the other methods.

All programs were written in FORTRAN and run on the USNA/DTSS computer system. All data and charts included in the paper.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED.

Security Classification

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

Security Classification